AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

1 of 6

ASL-TR-0107

AD

Reports Control Symbol
OSD 1366

SUPPLEMENT TO EOSAEL 80
VOLUME II
USER'S MANUAL

PROGRAM LISTINGS FOR EOSAEL 80-B
AND ANCILLARY CODES AGAUS AND FLASH

FEBRUARY 1982

By

R. G. Steinhoff

Approved for public release; distribution unlimited.

US Army Electronics Research and Development Command
Atmospheric Sciences Laboratory
White Sands Missile Range, NM 88002

82 05 11 016

NOTICES

Disclaimers

Disposition

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER ASL-TR-0107	2. GOVT ACCESSION NO. AD-A114417	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SUPPLEMENT TO EOSAEL 80 Vol II USER'S MANUAL PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND FLASH		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R. G. Steinhoff		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Atmospheric Sciences Laboratory White Sands Missile Range, NM		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Task AH71612111 H710011
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Electronics Research and Development Command Adelphi, MD 20783		12. REPORT DATE February 1982
		13. NUMBER OF PAGES 510
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DTIC COPY INSPECTED 2

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

EOSAEL 80
Computer Codes
Transmission
Transport and diffusion

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Computer listing of the Electro-Optical Systems Atmospheric Effects Library, version 80-B.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

PRECEDING PAGE BLANK-NOT FILMED

INTRODUCTION

This listing of EOSAEL 80-B is a supplement to Volume II[1] and supersedes all previous listings.[2] The current listing is complete as of 8 February 1982 and has revisions one through five incorporated into it.

EOSAEL 80-B differs from EOSAEL 80 in that modules SPOT, LT4M, NMMW, CLIMAT, BASCAT, and TURB have been extensively revised and, therefore, appear with new sequence numbers. All other modules have their original sequencing, except where revisions have been inserted or deleted.

The programs are listed by module with each module followed by its subroutines. Subroutines that have been listed for prior modules in the listing are not repeated in the source listing. The table of contents lists each module along with all its corresponding subroutines and the page number of each subroutine in the listing. The elements EOMAIN, COMPLT, and RESET, which are always to be resident, appear only at the beginning of the table of contents and the source listing.

Also included herein is a sample input file, NEWRUN, and an output file, EOOUT, produced by using the aforementioned sample input file.

The supplemental codes AGAUS and FLASH are supplied with EOSAEL 80. FLASH is described in appendix A of volume II of the User's Manual[1] and is further described in the comments of the source listing. Operating instructions for the AGAUS code may be found in the comments of the source listing. Manuals for AGAUS are available upon written request from the US Army Atmospheric Sciences Laboratory, White Sands Missile Range, New Mexico.

---

[1]Shirkey, R. C., and S. G. O'Brien, EOSAEL 80, Volume II, User's Manual, ASL-TR-0073, US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM, 1981.

[2]Steinhoff, R. G., Program Listings for EOSAEL 80 and Ancillary Codes AGAUS and FLASH, ASL-TR-0073 (Supplement), US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM, 1981.

```
      PROGRAM EOSAEL
C   MAIN PROGRAM FOR EOSAEL 80                                          EOM00010
C                                                                       EOM00020
      REAL LOTRNS,LAZTRN,LZTRN,MMTRAN,MMWTRN,IPNAM,IAL,IALB1,IALB2      EOM00030
      LOGICAL ISPOT,N16,LOREAD                                          EOM00040
      COMMON /SPOTLO/ISPOT,LOREAD,N16                                   EOM00050
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     EOM00060
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,      EOM00070
     1       WNDVEL,WNDDIR,IPASCT                                       EOM00080
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUEOM00090
      COMMON /GEOMET/PTS(15),IGEOSW                                     EOM00100
C           IOIN - CARD READER                                          EOM00110
C           IOOUT - PRINTER                                             EOM00120
C           IPHFUN - UNIT UPON WHICH PHASE FUNCTION DATA RESIDES        EOM00130
C           LOUNIT - UNIT UPON WHICH LT4M ATM DATA RESIDES              EOM00140
C           NDIRTU - UNIT UPON WHICH DRTRAN DATA RESIDES                EOM00150
C           NCLIMT - UNIT FOR CLIMATOLOGICAL DATA                       EOM00160
C           KSTOR  - AUXILLARY START/RESTART UNIT FOR BASCAT            EOM00170
C           NPLOTU - OPTIONAL UNIT FOR WRITING RESULTS FOR SUBSEQUENT
C                    PLOTTING PURPOSES BY THE USER.
      DIMENSION TRAN(16),RADA(16),RADG(16),IPROGN(20)                   EOM00180
      DIMENSION IDOPGM(20),IPNAM(40)                                    EOM00190
      DIMENSION IAL(12),DAT(10)                                         EOM00200
C     FOR UNIVAC                                                        EOM00210
      DATA IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR/                EOM00220
C    1      5,6,3,8,7,24,25/                                            EOM00230
C     PROGRAM NAMES                                                     EOM00240
      DATA IPNAM/4HSPOT,4HTURB,4HBASC,4HLT4M,4HXSCA,4HSMOK,4HDRTR,      EOM00250
     1      4HLZTR,4HNMMW,4HCLTR,4HSCRE,4HFCLO,4HOVRC,4HGRNA,4H*    ,   EOM00260
     2      4H*   ,4H*   ,4H*   ,4H*   ,4HCLIM,4H    ,4H    ,4HAT   ,   EOM00270
     3      4H    ,4HLE  ,4HE   ,4HAN  ,4HAN  ,4H    ,4HAN  ,4HEN   ,   EOM00280
     4      4HUD  ,4HST  ,4HDE  ,4H    ,4H    ,4H    ,4H    ,4H     ,   EOM00290
     5      4HATE /                                                     EOM00300
C     CARD MNEMONICS                                                    EOM00310
      DATA IAL/4HEORU,4HVIS ,4HFREQ,4HWAVL,4HWVNU,4HRESF,               EOM00320
     14HTARG,4HRCVR,4HDESG,4HOBSV,4HBFCL,4HGO  /                        EOM00330
      DATA PI,TORRMB,CDEGK/3.14159265,1.33322,273.16/                   EOM00340
      DATA PTS/15*0.0/                                                  EOM00350
      ISTART=0                                                          EOM00360
      CLDAMT=0.                                                         EOM00370
      CLDHYT=0.                                                         EOM00380
      FOGPRB=0.                                                         EOM00390
      PI2=PI/2.                                                         EOM00400
      PIRAD=PI/180.                                                     EOM00410
      TWOPI=2.0*PI                                                      EOM00420
      WRITE (IOOUT,1060)                                                EOM00430
C************************* I/O *****************************************EOM00440
C                                                                       EOM00450
C** INPUT TO EOSAEL IS CARD ORDER-INDEPENDENT, WITH EACH INPUT RECORD   EOM00460
C** HAVING A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4. THE ONLY EXCEPTION  EOM00470
C** TO THIS RULE IS THE GO SENTINEL CARD, WHICH MUST BE THE LAST RECORD EOM00480
C** IN THE INPUT SEQUENCE. ALL RECORDS ARE READ IN UNDER THE            EOM00490
C** FORMAT (2A4,1X,10E7.4). INTEGERS MUST BE INPUT AS REAL              EOM00500
C** NUMBERS IN THIS COMMON FORMAT SCHEME. THEY ARE LATER FIXED TO       EOM00510
C** THE INTEGER TYPE. THE IDENTIFIERS FOR EACH OF THE INPUT             EOM00520
C** RECORDS ARE AS FOLLOWS :                                            EOM00530
C                                                                       EOM00540
C---------------------------------------------------------------        EOM00550
C     CARD IDENTIFIER : EORUN                                           EOM00560
C     VARIABLES READ : NUMRUN                                           EOM00570
C      NUMRUN - NUMBER OF TIMES EOSAEL DRIVER IS TO BE CYCLED           EOM00580
C               DEFAULT IS 1.                                           EOM00590
C---------------------------------------------------------------        EOM00600
C     CARD IDENTIFIER : VIS                                             EOM00610
C     VARIABLES READ : VIS,EXTN55,EXTN                                  EOM00620
C      VIS - VISIBILITY AT WAVELENGTH OF 0.55 MICRONS (KM)              EOM00630
C      EXTN55 - EXTINCTION COEFFICIENT AT 0.55 MICRONS (KM**-1)         EOM00640
C      EXTN - EXTINCTION COEFFICIENT AT INPUT WAVELENGTH (KM**-1)       EOM00650
C    ** NOTE : IF THE VIS CARD IS NOT INPUT, A WARNING IS PRINTED       EOM00660
C               AND THE VISIBILITY IS SET TO A DEFAULT VALUE OF 10 KM.  EOM00670
```

```
C     ** NOTE : IF EXTN55 IS INPUT AS A VALUE LESS THAN 0.0001, IT
C                IS SET EQUAL TO THE QUOTIENT 3.912/VIS. IF VIS IS
C                INPUT AS A VALUE LESS THAN 0.0001, IT IS SET EQUAL TO
C                THE QUOTIENT 3.912/EXTN55.
C      ** NOTE : EXTN IS NEEDED ONLY FOR BASCAT
C-------------------------------------------------------------------
C
C** ONLY ONE OF THE FOLLOWING THREE CARDS MAY BE INPUT FOR A GIVEN
C** CYCLE OF EOSAEL. IF NONE OF THESE CARDS IS PRESENT, AN ERROR
C** MESSAGE IS PRINTED AND EXECUTION IS TERMINATED.
C
C
C      CARD IDENTIFIER : FREQ
C      VARIABLES READ : FREQ1, FREQ2, MULDV
C       FREQ1 - LOWER INPUT FREQUENCY (GHZ)
C       FREQ2 - HIGHER INPUT FREQUENCY (GHZ)
C       MULDV - FREQUENCY INCREMENT FOR SPOT AND/OR LT4M (GHZ)
C
C      CARD IDENTIFIER : WAVL
C      VARIABLES READ : WAVE1, WAVE2, MULDV
C       WAVE1 - SHORTER INPUT WAVELENGTH (MICRONS)
C       WAVE2 - LONGER INPUT WAVELENGTH (MICRONS)
C       MULDV - WAVELENGTH INCREMENT FOR SPOT AND/OR LT4M (MICRONS)
C
C      CARD IDENTIFIER : WVNUM
C      VARIABLES READ : WVNUM1, WVNUM2, MULDV
C       WVNUM1 - LOWER INPUT WAVENUMBER (CM**-1)
C       WVNUM2 - HIGHER INPUT WAVENUMBER (CM**-1)
C       MULDV - WAVENUMBER INCREMENT FOR SPOT AND/OR LT4M (CM**-1)
C-------------------------------------------------------------------
C
C** THE NEXT CARD DETERMINES WHETHER A SENSOR RESPONSE FUNCTION
C** IS DESIRED FOR BROAD BAND CALCULATIONS. THIS OPTION IS
C** INVOKED ONLY IF THIS CARD IS PRESENT.
C
C      CARD IDENTIFIER : RESF
C      VARIABLES READ : NONE HERE - SEE SPOT OR LT4M WRITEUP FOR
C                    PROPER PLACEMENT OF RESPONSE FUN CARDS.
C-------------------------------------------------------------------
C
C** THE NEXT FIVE CARDS COMPRISE THE GEOMETRICAL OPTION OF EOSAEL.
C** THIS OPTION IS USEFUL FOR EOSAEL RUNS WHERE SEVERAL MODULES
C** EXAMINE DIFFERENT ATMOSPHERIC OBSCURATION EFFECTS ALONG THE
C** SAME PHYSICAL PATH. THE GEOMETRICAL OPTION ASSURES THAT THE
C** POINTS OF REFERENCE IN THE SCENARIO UNDER EXAMINATION ARE
C** CONSISTENTLY SPECIFIED FOR ALL MODULES. IT SHOULD BE NOTED
C** THAT THIS OPTION IS ACTIVATED WHENEVER ANY OF THE FIVE CARDS IS
C** ENCOUNTERED. ONCE THE OPTION IS ACTIVATED IT IS IMPORTANT THAT
C** AT LEAST THE FIRST TWO CARDS (TARG AND RCVR) BE INPUT TO
C** DEFINE THE PHYSICAL PATH, SINCE THIS OPTION WILL OVERRIDE
C** POSITIONS OR LENGTHS CONTAINED IN NORMAL INPUT TO ALL MODULES.
C** THE GEOMETRICAL INPUT CONSISTS OF FIVE SETS OF COORDINATES
C** WHICH OBEY THE FOLLOWING CONVENTIONS :
C**        (A) ALL COORDINATES ARE DIMENSIONED IN KILOMETERS
C          (B) THE Z-AXIS IS POSITIVE UPWARD
C          (C) THE Y-AXIS POINTS NORTH
C          (D) THE X-AXIS POINTS EAST
C
C** THE FIVE GEOMETRICAL CARDS ARE AS FOLLOWS :
C
C
C      CARD IDENTIFIER : TARG
C      VARIABLES READ : PTS(1),PTS(2),PTS(3)
C       PTS(1-3) - COORDINATES OF THE TARGET (FOR THE DRTRAN MODULE,
C                  THESE ARE THE COORDINATES OF THE TRANSMITTER).
C
C      CARD IDENTIFIER : RCVR
C      VARIABLES READ : PTS(4),PTS(5),PTS(6)
C       PTS(4-6) - COORDINATES OF THE RECEIVER OR SEEKER
C
```

```
C       CARD IDENTIFIER : DESG                                          EOM01380
C       VARIABLES READ : PTS(7),PTS(8),PTS(9)                           EOM01390
C        PTS(7-9) - COORDINATES OF THE DESIGNATOR OR SOURCE             EOM01400
C                                                                       EOM01410
C       CARD IDENTIFIER : OBSV                                          EOM01420
C       VARIABLES READ : PTS(10),PTS(11),PTS(12)                        EOM01430
C        PTS(10-12) - COORDINATES OF THE OBSERVER USED BY DRTRAN        EOM01440
C                                                                       EOM01450
C       CARD IDENTIFIER : BFCL                                          EOM01460
C       VARIABLES READ : PTS(13),PTS(14),PTS(15)                        EOM01470
C        PTS(13-15) - COORDINATES OF THE CENTER OF THE CLOUD ELLIPSOID  EOM01480
C                     USED BY BASCAT AND FCLOUD                         EOM01490
C-------------------------------------------------------------------    EOM01500
C                                                                       EOM01510
C** THE NEXT INPUT CARD IS THE CLIMATOLOGICAL OPTION CARD. THIS         EOM01520
C** OPTION ALLOWS USER INPUT OF METEOROLOGICAL PARAMETERS DIRECTLY       EOM01530
C** OR AUTOMATIC INPUT OF CLIMATOLOGY DATA CHARACTERISTIC OF            EOM01540
C** WEST GERMANY.  IF THIS OPTION IS INVOKED ALL MODULES WILL USE       EOM01550
C** THIS DATA, I.E. MET DATA THAT HAS BEEN INPUT TO A SPECIFIC          EOM01560
C** MODULE WILL BE OVERRIDDEN.                                          EOM01570
C                                                                       EOM01580
C                                                                       EOM01590
C       CARD IDENTIFIER : CLIMAT                                        EOM01600
C       VARIABLES READ : ICLMAT,LOCAT,MONTH,NHOUR,IWIND,NPRT            EOM01610
C         ** OR **        ICLMAT,IPASCT,TEMP,PRESS,RH,AH,DP,VIS,WNDVEL, EOM01620
C                   WINDDIR                                             EOM01630
C           LOCAT - CLIMATOLOGY REGION INDICATOR. LOCAT IS AN INTEGER
C                    (1-4) FOR CENTRAL EUROPE AND
C                    (5-10) FOR MID-EAST.
C
C                   L = 1 - EUROPEAN LOWLANDS,
C                   L = 2 - EUROPEAN RHINE VALLEY,
C                   L = 3 - EUROPEAN HIGHLANDS,
C                   L = 4 - EUROPEAN ALPINE,
C                   L = 5 - MIDEAST DESERTS,
C                   L = 6 - MIDEAST COASTAL,
C                   L = 7 - MIDEAST PERSIAN GULF,
C                   L = 8 - MIDEAST RED SEA,
C                   L = 9 - MIDEAST EASTERN MOUNTAINS, AND
C                   L = 10 - MIDEAST INDUS VALLEY.
C
C           MONTH - AN INTEGER (1-12) INDICATING THE MONTH OF THE YEAR.
C                    MONTH IS USED TO SELECT THE SEASON WHICH IS
C                    APPLICABLE TO THE REGION LOCAT.
C           NHOUR - AN INTEGER (0-23) INDICATING THE TIME OF DAY LOCAL
C                    STANDARD TIME (LST). NHOUR IS USED TO SELECT ONE OF
C                    FOUR TIME PERIODS OF THE DAY 20-02, 03-09, 10-14,
C                    AND 15-19.
C           IWIND - *** NOT USED ***
C           NPRT - A PRINT SELECTOR.
C                   NPRT LE ZERO - DO NOT PRINT CLIMATOLOGICAL DATA.
C                   NPRT GT ZERO - PRINT ALL AVAILABLE MEANS, STANDARD
C                     DEVIATIONS, AND PERCENT OCCURRENCES.
C         ICLMAT = 2.: USER INPUT QUANTITIES                            EOM01820
C           IPASCT = PASQUILL STABILITY CATEGORY VALID RANGE =1.-6.(A-F)EOM01830
C           TEMP = TEMPERATURE IN DEGREES C                             EOM01840
C           PRESS = PRESSURE IN MB (SEA LEVEL IF ICLMAT=1)              EOM01850
C           RH = RELATIVE HUMIDITY IN %                                 EOM01860
C           AH = ABSOLUTE HUMIDITY - DEFINED HERE AS THE H2O VAPOR      EOM01870
C                DENSITY IN G/M**3.                                     EOM01880
C           DP = DEW POINT TEMPERATURE IN DEGREES C                     EOM01890
C           VIS = VISIBILITY IN KM                                      EOM01900
C           WNDVEL = WIND VELOCITY IN M/S (DEPNT UPON IWIND IF ICLMAT=1)EOM01910
C           WINDIR = WIND DIRECTION IN DEGREES (IF ICLMAT=1 WILL BE     EOM01920
C                                   MOST PROBABLE DIRECTION)            EOM01930
C-------------------------------------------------------------------    EOM01940
C                                                                       EOM01950
C** THE FOLLOWING CARDS ARE ALSO READ IN UNDER THE COMMON FORMAT        EOM01960
C** USED ABOVE. THE INFORMATION ON EACH OF THESE RECORDS DETERMINES     EOM01970
C** WHICH MODULES ARE SELECTED AND HOW MANY TIMES THE MODULES ARE       EOM01980
```

```
C** TO BE CYCLED WITHIN EACH CYCLE OF THE EOSAEL DRIVER.
C
C
C     CARD IDENTIFIER : (SEE MODULE IDENTIFIERS BELOW)
C     VARIABLES READ : IDOPGM
C      IDOPGM - NUMBER OF TIMES THE SELECTED MODULE IS TO
C               BE CYCLED WITHIN EACH EOSAEL CYCLE - DEFAULT IS ONE.
C
C
C            NO.  MODULE IDENTIFIER      --------- RANGES -------------
C            ---  -----------------       WAVE(UM)            FREQ(GHZ)
C             1         SPOT        .25-2.,3.-5.,8.-12.  *
C             2         TURB                LT 14.       *
C             3         BASCAT     ANY WAVELENGTH IN DATA FILE IPHFUN
C             4         LT4M       .25-2.,3.-5.,8.-12.   *
C             5         XSCALE          1.06,3-5,8-12    *
C             6         SMOKE          .4-1.2,3-5,8-12   *         94.
C             7         DRTRAN     .4-1.1,3.5-4.,8.5-12.*       94.-140.
C             8         LZTRAN            .8-11.         *
C             9         NMMW                             *        10-350
C            10         CLTRAN        .2-2.,3.-5.,8.-12.*
C            11         SCREEN        N/A
C            12         FCLOUD     ANY WAVELENGTH IN DATA FILE IPHFUN
C            13         OVRCST           ANY WAVELENGTH
C            14         GRNADE        SAME AS SMOKE
C**  NOTE : THE DATA SPECIFIC TO EACH MODULE MUST BE INPUT IN
C     THE SEQUENCE IN THE ABOVE LIST.
C--------------------------------------------------------------------
C     CARD IDENTIFIER : GO
C     VARIABLES READ : NONE
C      END OF READ SENTINEL (MUST BE LAST CARD READ).
C
C***********************************************************************
      NUMRUN=1
      IRFLAG=0
      READ (IOIN,1000) IALB1,IALB2,(DAT(L),L=1,10)
      IF (IALB1.NE.IAL(1)) GO TO 10
      NUMRUN=IFIX(DAT(1))
      IF (NUMRUN.EQ.0) NUMRUN=1
      GO TO 20
C     SET FLAG IF EORUN IS NOT THE FIRST CARD
   10 IRFLAG=1
   20 CONTINUE
      DO 580 JRUN=1,NUMRUN
C     INITIALIZATION
        DO 30 I=1,20
C     PROGRAM SELECTOR
        IPROGN(I)=0
C     PROGRAM CYCLE DEFAULT
   30   IDOPGM(I)=1
C     TRANSMISSIONS
      LOTRNS=1.
      XSTRN=1.
      SMKTRN=1.
      DRTRN=1.
      LZTRN=1.
      MMWTRN=1.
      CLTRN=1.0
      GRNTRN=1.0
      IF (JRUN.GT.1) WRITE (IOOUT,1070) JRUN
C     FREQUENCY, WAVELENGTH, WAVENUMBER INDICATOR
      IFW=0
C     GEOMETRICAL OPTION SWITCH
      IGEOSW=0
C     SENSOR RESPONSE FUNCTION OPTION SWITCH
      NR=0
      VIS=0.0
      EXTN55=0.0
      EXTN=0.0
        DO 220 J=1,25
```

```
      IF (IRFLAG.EQ.1) WRITE (IOOUT,1010)
C     SUPRESS READ IN CASE FIRST CARD PREVIOUSLY READ WASNT EORUN.
      IF (IRFLAG.EQ.0) READ (IOIN,1000) IALB1,IALB2,(DAT(L),L=1,10)
      IRFLAG=0
      INOPT=0
      IF (J.EQ.25) GO TO 230
        DO 40 KK=1,12
C     CHECK FOR CARD TYPES, NOT PROGRAM SELECTOR
        IF (IALB1.NE.IAL(KK)) GO TO 40
        INOPT=KK
        IF (INOPT.GE.3.AND.INOPT.LE.5) IFW=INOPT
C     GO CARD FOUND
        IF (INOPT.EQ.12) GO TO 250
        GO TO 80
   40   CONTINUE
C     SEARCH FOR PROGRAMS HERE
        DO 50 KK=1,20
        IF (IALB1.NE.IPNAM(KK)) GO TO 50
        IPROGN(KK)=KK
        IF (DAT(1).GT.1.0) IDOPGM(KK)=IFIX(DAT(1))
        IF (KK.EQ.20) GO TO 60
        GO TO 220
   50   CONTINUE
      GO TO 240
C     CLIMATOLOGICAL OPTION INVOKED
   60 ICLMAT=IFIX(DAT(1))
      IF (ICLMAT.EQ.2) GO TO 70
      LOCAT=IFIX(DAT(2))
      MONTH=IFIX(DAT(3))
      NHOUR=IFIX(DAT(4))
      IWIND=IFIX(DAT(5))
      NPRT=IFIX(DAT(6))
      GO TO 220
   70 IPASCT=IFIX(DAT(2))
      TEMP=DAT(3)
      PRESS=DAT(4)
      RH=DAT(5)
      AH=DAT(6)
      DP=DAT(7)
      VIS=DAT(8)
      WNDVEL=DAT(9)
      WNDDIR=DAT(10)
      GO TO 220
C     GEOMETRICAL OPTION INVOKED
   80 IF (INOPT.GT.6) GO TO 90
C     CARD SETUP SWITCHING
      GO TO (210,160,170,180,190,200),INOPT
   90 LPTSSW=INOPT-6
      IGEOSW=1
        DO 150 K=1,3
        GO TO (100,110,120,130,140),LPTSSW
  100   PTS(K)=DAT(K)
        GO TO 150
  110   PTS(K+3)=DAT(K)
        GO TO 150
  120   PTS(K+6)=DAT(K)
        GO TO 150
  130   PTS(K+9)=DAT(K)
        GO TO 150
  140   PTS(K+12)=DAT(K)
  150   CONTINUE
      GO TO 220
C     VISIBILITY CARD
  160 VIS=DAT(1)
      EXTN55=DAT(2)
      EXTN=DAT(3)
      GO TO 220
C     FREQUENCY CARD
  170 FREQ1=DAT(1)
      FREQ2=DAT(2)
```

```
      MULDV=IFIX(DAT(3))                                                 EOM03390
      GO TO 220                                                          EOM03400
C     WAVELENGTH CARD                                                    EOM03410
  180 WAVE1=DAT(1)                                                       EOM03420
      WAVE2=DAT(2)                                                       EOM03430
      MULDV=IFIX(DAT(3))                                                 EOM03440
      GO TO 220                                                          EOM03450
C     WAVENUMBER CARD                                                    EOM03460
  190 WVNUM1=DAT(1)                                                      EOM03470
      WVNUM2=DAT(2)                                                      EOM03480
      MULDV=IFIX(DAT(3))                                                 EOM03490
      GO TO 220                                                          EOM03500
C     SENSOR OPTION INVOKED                                              EOM03510
  200 NR=1                                                               EOM03520
      GO TO 220                                                          EOM03530
  210 WRITE (IOOUT,1010)                                                 EOM03540
  220 CONTINUE                                                           EOM03550
      IF (IFW.NE.0) GO TO 250                                            EOM03560
C     ERROR CHECK ON WAVENUMBER, WAVELENGTH, OR FREQUENCY                EOM03570
      WRITE (IOOUT,1020)                                                 EOM03580
      GO TO 580                                                          EOM03590
  230 WRITE (IOOUT,1030)                                                 EOM03600
      GO TO 250                                                          EOM03610
C     UNKNOWN CARD TYPE                                                  EOM03620
  240 WRITE (IOOUT,1040) IALB1,IALB2                                     EOM03630
  250 CONTINUE                                                           EOM03640
C     SELECT FREQUENCY, WAVELENGTH, OR WAVENUMBER                        EOM03650
      IF (IFW-4) 260,270,280                                             EOM03660
  260 WVNUM1=FREQ1/30.                                                   EOM03670
      WVNUM2=FREQ2/30.                                                   EOM03680
      WAVE1=0.                                                           EOM03690
      IF (FREQ2.GT..0001) WAVE1=3.E+05/FREQ2                             EOM03700
      WAVE2=3.E+05/FREQ1                                                 EOM03710
      GO TO 290                                                          EOM03720
  270 WVNUM1=0.                                                          EOM03730
      IF (WAVE2.GT..0001) WVNUM1=1.E+04/WAVE2                            EOM03740
      WVNUM2=1.E+04/WAVE1                                                EOM03750
      FREQ1=30.*WVNUM1                                                   EOM03760
      FREQ2=30.*WVNUM2                                                   EOM03770
      GO TO 290                                                          EOM03780
  280 FREQ1=30.*WVNUM1                                                   EOM03790
      FREQ2=30.*WVNUM2                                                   EOM03800
      WAVE1=0.                                                           EOM03810
      IF (WVNUM2.GT..0001) WAVE1=1.E+04/WVNUM2                           EOM03820
      WAVE2=1.E+04/WVNUM1                                                EOM03830
  290 CONTINUE                                                           EOM03840
      IF (VIS.LT..0001.AND.EXTN55.LT.0001) WRITE (IOOUT,1050)            EOM03850
      IF (VIS.LT..0001.AND.EXTN55.LT.0001) VIS=10.                       EOM03860
      IF (EXTN55.GT..0001) VIS=3.912/EXTN55                              EOM03870
      IF (VIS.GT..0001) EXTN55=3.912/VIS                                 EOM03880
C     OUTPUT INFORMATION                                                 EOM03890
      WRITE (IOOUT,1080)                                                 EOM03900
      DO 300 I=1,20                                                      EOM03910
  300 IF (IPROGN(I).EQ.I) WRITE (IOOUT,1090) IPNAM(I),IPNAM(I+20)        EOM03920
      WRITE (IOOUT,1100) WVNUM1,WVNUM2,WAVE1,WAVE2,FREQ1,FREQ2           EOM03930
C     CLIMAT USES UNIT NCLIMT                                            EOM03940
      IF (ICLMAT.EQ.1) CALL CLIMAT(LOCAT,MONTH,NHOUR,IWIND,NPRT,TEMP,    EOM03950
     1    PRESS,RH,AH,DP,VIS,WNDVEL,WNDDIR,IPASCT)                       EOM03960
      WRITE (IOOUT,1110) VIS                                             EOM03970
      IF (ICLMAT.EQ.2) ICLMAT=1                                          EOM03980
C***************** SPOT CONTRAST PGM *********************************EOM03990
      IF (IPROGN(1).NE.1) GO TO 320                                      EOM04000
      IPGM1=IDOPGM(1)                                                    EOM04010
      WRITE (IOOUT,1130)                                                 EOM04020
      DO 310 I=1,IPGM1                                                   EOM04030
C     SPOT USES UNITS: IPHFUN - PHASE FUNCTION; LOUNIT - LT4M DATA       EOM04040
  310 CALL SPOT(WVNUM1,WVNUM2,VIS,NR,IERR,MULDV)                         EOM04050
      CALL RESET(IERR)                                                   EOM04060
C****************** TURBULENCE PGM **********************************EOM04070
  320 IF (IPROGN(2).NE.2) GO TO 340                                      EOM04080
```

```
      IPGM2=IDOPGM(2)
      WRITE (IOOUT,1140)
        DO 330 I=1,IPGM2
  330   CALL TURB(WAVE1,IERR)
      CALL RESET(IERR)
C****************** LASER MULTIPLE SCATTERING PGM ********************
  340 IF (IPROGN(3).NE.3) GO TO 360
      IPGM3=IDOPGM(3)
      ISPOT=.FALSE.
      WRITE (IOOUT,1150)
        DO 350 I=1,IPGM3
C     BASCAT USES UNIT IPHFUN FOR PHASE FUNCTION DATA
  350   CALL BASCAT(WAVE1,EXTN,IERR)
      CALL RESET(IERR)
C****************** LT4M PGM ****************************************
  360 IF (IPROGN(4).NE.4) GO TO 380
C     LT4M READS ATM DATA FROM LOUNIT
      ISPOT=.FALSE.
      LOREAD=.TRUE.
      IPGM4=IDOPGM(4)
        DO 370 I=1,IPGM4
        CALL LT4M(H1,H2,ANGLE,ITYPE,IXY,TRAN,RADA,RADG,IEMISS,LEN,MODEL,
     1       VIS,WVNUM1,WVNUM2,T1,ICLMAT,IERR,NR,IHAZE,MULDV)
  370   LOTRNS=LOTRNS*TRAN(1)
      CALL RESET(IERR)
C****************** XSCALE EXTINCTION PGM ***************************
  380 IF (IPROGN(5).NE.5) GO TO 400
      WRITE (IOOUT,1160)
      IPGM5=IDOPGM(5)
        DO 390 I=1,IPGM5
        CALL XSCALE(WAVE1,VIS,EXTN55,XTRN,IERR,0,0,0.,0.)
  390   XSTRN=XSTRN*XTRN
      CALL RESET(IERR)
C****************** SMOKE PGM ***************************************
  400 IF (IPROGN(6).NE.6) GO TO 420
      WRITE (IOOUT,1170)
      IPGM6=IDOPGM(6)
        DO 410 N=1,IPGM6
        CALL SMOKE(WAVE1,ICLMAT,STRANS,IERR)
  410   SMKTRN=SMKTRN*STRANS
      CALL RESET(IERR)
C****************** DRTRAN  PGM *************************************
  420 IF (IPROGN(7).NE.7) GO TO 440
      WRITE (IOOUT,1180)
      IPGM7=IDOPGM(7)
      HOLDWV=WAVE1
      IF (IFW .EQ.1) WAVE1=WAVE2
        DO 430 N=1,IPGM7
C      DRTRAN USES NDIRTU FOR DATA
        IF (N.GT.1) WRITE (IOOUT,1120)
        CALL DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)
  430   DRTRN=DRTRN*TRNLOS
      WAVE1=HOLDWV
      CALL RESET(IERR)
C****************** LASER TRANSMISSION PGM **************************
  440 IF (IPROGN(8).NE.8) GO TO 460
      WRITE (IOOUT,1190)
      IPGM8=IDOPGM(8)
        DO 450 I=1,IPGM8
        CALL LZTRAN(WAVE1,ICLMAT,LAZTRN,IERR)
  450   LZTRN=LZTRN*LAZTRN
      CALL RESET(IERR)
C****************** NEAR MILLIMETER WAVE PGM ************************
  460 IF (IPROGN(9).NE.9) GO TO 480
      WRITE (IOOUT,1200)
      IPGM9=IDOPGM(9)
        DO 470 I=1,IPGM9
        CALL NMMW(FREQ1,ICLMAT,MMTRAN,IERR)
  470   MMWTRN=MMWTRN*MMTRAN
      CALL RESET(IERR)
```

```
C***************** CLOUD TRANSMISSION PGM ******************************EOM04790
  480 IF (IPROGN(10).NE.10) GO TO 500                                   EOM04800
      IPGM10=IDOPGM(10)                                                 EOM04810
        DO 490 I=1,IPGM10                                               EOM04820
        WRITE (IOOUT,1210)                                              EOM04830
        CALL CLTRAN(CTRANS,WAVE1,I,IERR)                                EOM04840
  490   CLTRN=CLTRN*CTRANS                                              EOM04850
      CALL RESET(IERR)                                                  EOM04860
C**** CWIC MUNITION EXPENDITURES/INVERSE STATIC TARGET DETECTION PGM ***EOM04870
  500 IF (IPROGN(11).NE.11) GO TO 520                                   EOM04880
      WRITE (IOOUT,1220)                                                EOM04890
      IPGM11=IDOPGM(11)                                                 EOM04900
        DO 510 I=1,IPGM11                                               EOM04910
  510   CALL SCREEN(IERR,ICLMAT)                                        EOM04920
      CALL RESET(IERR)                                                  EOM04930
C*************** FINITE CLOUD RADIATIVE TRANSFER PGM *****************EOM04940
  520 IF (IPROGN(12).NE.12) GO TO 540                                   EOM04950
      WRITE (IOOUT,1230)                                                EOM04960
      IPGM12=IDOPGM(12)                                                 EOM04970
C     FCLOUD USES IPHFUN FOR PHASE FUNCTION DATA                        EOM04980
        DO 530 I=1,IPGM12                                               EOM04990
  530   CALL FCLOUD(WAVE1,FTRANS,IERR)                                  EOM05000
      CALL RESET(IERR)                                                  EOM05010
C************** OVERCAST SKY RADIATIVE TRANSFER PGM ***************** EOM05020
  540 IF (IPROGN(13).NE.13) GO TO 560                                   EOM05030
      WRITE (IOOUT,1240)                                                EOM05040
      IPGM13=IDOPGM(13)                                                 EOM05050
        DO 550 I=1,IPGM13                                               EOM05060
  550   CALL OVRCST(WAVE1,OTRANS,IERR)                                  EOM05070
      CALL RESET(IERR)                                                  EOM05080
C************ SELF-SCREENING SMOKE GRENADE PGM ********************** EOM05090
  560 IF (IPROGN(14).NE.14) GO TO 575                                   EOM05100
      WRITE (IOOUT,1250)                                                EOM05110
      IPGM14=IDOPGM(14)                                                 EOM05120
        DO 570 I=1,IPGM14                                               EOM05130
        CALL GRNADE(WAVE1,ICLMAT,GRTRAN,IERR)                           EOM05140
  570   GRNTRN=GRNTRN*GRTRAN                                            EOM05150
      CALL RESET(IERR)                                                  EOM05160
C******************************************************************** EOM05170
  575 IF(IPROGN(4).GT.0.OR.IPROGN(5).GT.0.OR.IPROGN(6).GT.0.OR.IPROGN(7)
     + .GT.0.OR.IPROGN(8).GT.0.OR.IPROGN(9).GT.0.OR.IPROGN(10).GT.0.OR.
     + IPROGN(14).GT.0) GO TO 576
      GO TO 580
  576 CALL COMPLT(LOTRNS,XSTRN,SMKTRN,DRTRN,LZTRN,MMWTRN,GRNTRN,CLTRN)  EOM05180
  580 CONTINUE                                                          EOM05190
      WRITE (IOOUT,1260)                                                EOM05200
      STOP                                                              EOM05210
C                                                                       EOM05220
C                                                                       EOM05230
 1000 FORMAT(2A4,1X,10E7.4)                                             EOM05240
 1010 FORMAT(1H0,20X,75H***EOMAIN WARNING*** EORUN CYCLE CARD OUT OF SEQEOM05250
     1UENCE, DEFAULT TO ONE CYCLE //)                                   EOM05260
 1020 FORMAT(1H0,20X,74H***EOMAIN ERROR*** FREQ, WAVL, OR WVNUM CARD WASEOM05270
     1 NOT INPUT, RUN TERMINATED //)                                    EOM05280
 1030 FORMAT(1H0,20X,46H***EOMAIN ERROR*** END OF READ SENTINEL ABSENT /EOM05290
     1/,1X,20X,28HRESULTS MAY BE UNPREDICTABLE)                         EOM05300
 1040 FORMAT(1H0,20X,80H***EOMAIN ERROR*** INPUT CARD DETECTED WHICH DOEEOM05310
     1S NOT MATCH CORRECT INPUT FORMAT//,1X,20X,13HTHE CARD WAS:,2X,2A4)EOM05320
 1050 FORMAT (1H0,20X,24H**** EOSAEL WARNING ****,/,1X,20X,15HVISIBILITYEOM05330
     1 AND , 47HEXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM/)       EOM05340
 1060 FORMAT (1H1,///////,1X,50X,30(1H*),/,1X,50X,1H*,28X,1H*,/         EOM05350
     1          ,1X,50X,30H*    ELECTRO-OPTICAL SYSTEMS   *,/,1X,50X,1H*, EOM05360
     2          28X,1H*,/,1X,50X,30H*     ATMOSPHERIC EFFECTS      *,/  EOM05370
     3          ,1X,50X,1H*,28X,1H*,/,1X,50X,25H*         LIBRARY       EOM05380
     4          5H     *,/,1X,50X,1H*,28X,1H*,/,1X,50X,30(1H*))         EOM05390
 1070 FORMAT (1H1,/////,58X,11HRUN NUMBER ,I2)                          EOM05400
 1080 FORMAT (///1X,51X,28HINDIVIDUAL MODULES  SELECTED)                EOM05410
 1090 FORMAT (1X,62X,2A4)                                               EOM05420
 1100 FORMAT (1H0,63X,9HBEGINNING,12X,6HENDING,//,39X,14HWAVENUMBER(CM* EOM05430
     1          4H*-1),6X,F10.3,10X,F10.3,//,39X,19HWAVELENGTH(MICRONS), EOM05440
```

```
     2          5X,F10.3,10X,F10.3,//,39X,14HFREQUENCY(GHZ),5X,F15.3,5X,   EOM05450
     3          F15.3,//)                                                   EOM05460
1110  FORMAT (1H0,62X,10HVISIBILITY,/,62X,F5.2,3H KM)                       EOM05470
1120  FORMAT(1H1)                                                           EOM05480
1130  FORMAT(1H1,40X,20HSPOT CONTRAST MODULE //)                            EOM05490
1140  FORMAT(1H1,40X,17HTURB LASER MODULE //)                               EOM05500
1150  FORMAT(1H1,40X,30HBASCAT LASER SCATTERING MODULE //)                  EOM05510
1160  FORMAT (1H1,40X,46HXSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE     EOM05520
     1          ///)                                                        EOM05530
1170  FORMAT (1H1,45X,19HSMOKE MODEL MODULE ////)                           EOM05540
1180  FORMAT (1H1,40X,26H DIRT TRANSMISSION MODULE ///)                     EOM05550
1190  FORMAT (1H1,40X,28H LASER TRANSMITTANCE MODULE //)                    EOM05560
1200  FORMAT (1H1,45X,29H NEAR MILLIMETER WAVE MODULE ///)                  EOM05570
1210  FORMAT(1H1,40X,27HCLOUD TRANSMITTANCE MODULE ///)                     EOM05580
1220  FORMAT(1H1,20X,43HCWIC MUNITION EXPENDITURES / INVERSE STATIC         EOM05590
     1 24H TARGET DETECTION MODULE )                                        EOM05600
1230  FORMAT(1H1,40X,38HFINITE CLOUD RADIATIVE TRANSFER MODULE ///)         EOM05610
1240  FORMAT(1H1,40X,38HOVERCAST SKY RADIATIVE TRANSFER MODULE ///)         EOM05620
1250  FORMAT(1H1,40X,35HSELF-SCREENING SMOKE GRENADE MODULE ///)            EOM05630
1260  FORMAT (1X,/////,1X,50X,14HEND EOSAEL RUN)                            EOM05640
      END                                                                   EOM05650
```

```
      SUBROUTINE RESET (IERR)                                           RES00010
C     THE PURPOSE OF THIS ROUTINE IS: (1) TO RESET THE SEQUENCING OF    RES00020
C     DATA CARDS DUE TO AN ERROR IN A PREVIOUS MODULE OR (2) TO READ    RES00030
C     A SENTENAL CARD THAT DELINEATES THE END OF A DATA SET (SEE        RES00040
C     BELOW FOR DEFINITION OF A DATA SET) OR (3) TO STOP THE PROGRAM -  RES00050
C     THIS LAST MODE IS USALLY FOR DEBUGGING OR TO ONLY CHANGE A CARD   RES00060
C     IN A COMPLETE RUN.                                                RES00070
C                                                                       RES00080
C     TO DELINEATE THE END OF A DATA SET A CARD THAT HAS JUST END       RES00090
C     ON IT MUST BE INSERTED AS A SENTINEL CARD; A DATA SET IS          RES00100
C     DEFINED AS THAT COMPLETE SET OF CARDS NECESSARY TO RUN THE        RES00110
C     CALLED MODULE THE NUMBER OF TIMES AS SPECIFIED ON THE IDOPGM(I)   RES00120
C     CARD. STOP MAY ALSO BE INSERTED AS A SENTINAL CARD, IN WHICH      RES00130
C     CASE THE PROGRAM WILL BE TERMINATED AT THAT POINT - THIS IS       RES00140
C     NOT THE NORMAL TERMINATION.                                       RES00150
C                                                                       RES00160
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTURES00170
      DATA ICHCK1,ICHCK2,ISTOP1,ISTOP2 /2HEN,2HD ,2HST,2HOP/            RES00180
      IF(IERR.EQ.1) GO TO 1                                             RES00190
5     CONTINUE                                                          RES00200
C     FOR UNIVAC AND IBM                                                RES00210
C     READ (IOIN,100,END=2) ISNTL1,ISNTL2                               RES00220
      READ (IOIN,100) ISNTL1,ISNTL2                                     RES00230
C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE            RES00240
C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.            RES00250
C3    READ(IOIN,100) ISNTL1,ISNTL2                                      RES00260
C4    IF(EOF(IOIN)) 2,10                                                RES00270
 10   IF((ISNTL1.NE.ICHCK1.AND.ISNTL2.NE.ICHCK2).AND.                   RES00280
     1      (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 5            RES00290
      IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP                   RES00300
      RETURN                                                            RES00310
1     WRITE(IOOUT,102)                                                  RES00320
C     FOR UNIVAC AND IBM                                                RES00330
C6     READ (IOIN,100,END=2) ISNTL1,ISNTL2                              RES00340
6     READ (IOIN,100) ISNTL1,ISNTL2                                     RES00350
C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE            RES00360
C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.            RES00370
C3    READ(IOIN,100) ISNTL1,ISNTL2                                      RES00380
C4    IF(EOF(IOIN)) 2,20                                                RES00390
20    CONTINUE                                                          RES00400
      IF ((ISNTL1.NE.ICHCK1.AND.ISENTL.NE.ICHCK2).AND.                  RES00410
     1  (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 6                RES00420
      IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP                   RES00430
      IERR=0                                                            RES00440
      RETURN                                                            RES00450
2     WRITE (IOOUT,101) IOIN                                            RES00460
100   FORMAT (2A2)                                                      RES00470
101   FORMAT (1X,120(1H*),/,1X,29H ERROR IN INPUT CONTROL FILE ,I4,     RES00480
     + 21H - PROGRAM TERMINATED,/,1X,120(1H*))                          RES00490
102   FORMAT(1H0,50H**** CARD SEQUENCE RESET DUE TO ERROR IN PREVIOUS , RES00500
     1      15HMODULE (IERR=1)//)                                       RES00510
      STOP                                                              RES00520
      END                                                               RES00530
```

```
      SUBROUTINE COMPLT(LOTRNS,XSTRN,                                   COM00010
     +                  SMKTRN,DRTRN,LZTRN,MMWTRN,GRNTRN,CLTRN)         COM00020
      REAL LOTRNS,LZTRN,MMWTRN                                          COM00030
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCOM00040
C     LOWTRAN - LOTRNS                                                  COM00050
C      XSCALE - XSTRN                                                   COM00060
C     SMOKE - SMKTRN                                                    COM00070
C     DRTRAN - DRTRN                                                    COM00080
C     LZTRAN - LZTRN                                                    COM00090
C     NMMW - MMWTRN                                                     COM00100
C     CLTRAN - CLTRN                                                    COM00110
C     GRNADE - GRNTRN                                                   COM00120
      TRAN=LOTRNS*XSTRN*SMKTRN*DRTRN*CLTRN*GRNTRN*MMWTRN*LZTRN          COM00130
      WRITE (IOOUT,100) TRAN                                            COM00140
100   FORMAT (///1X,20X,24HCOMBINED TRANSMISSION FO,15HR THE SELECTED , COM00150
     1  10HMODULES = ,E10.4)                                            COM00160
      RETURN                                                            COM00170
      END                                                               COM00180
```

```
      SUBROUTINE ILLUM(LAMBDA,LD,E0)
      REAL LAMBDA,LUNPHA
      COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C*********************************************************************
C
C  SUBROUTINE ILLUM RETURNS THE EXTRATERRESTRIAL IRRADIANCE E0
C  AT WAVELENGTH LAMBDA.  IF LD = 0, THE VALUE GIVEN IS SOLAR
C  IRRADIANCE.  IF 1 < LD < 28 THE VALUE GIVEN IS LUNAR IRRAD-
C  IANCE ON LUNAR DAY LD, WITH DAY 28 CORRESPONDING TO FULL
C  MOON AND DAY 14 BEING NEW MOON.
C
C  SUBROUTINE COMPUTES VALUE OF LUNAR PHASE ANGLE, IF REQUIRED,
C  AND CALLS ONE OF THE EOSAEL ROUTINES SOLARS OR SMOON.
C
C*********************************************************************
      IF(ILD.GT.0) GO TO 10
      E0=SOLARS(LAMBDA)
      GO TO 100
C
C  LD GT 0 => E0 = LUNAR VALUE
C
   10 ILD=LD
      IF(ILD.GT.14) ILD=28-ILD
      LUNPHA=180.0*FLOAT(ILD)/14.0
      E0=SMOON(LAMBDA,LUNPHA)
  100 WRITE(IOOUT,1000) E0
      RETURN
 1000 FORMAT(32H0  EXTRATERRESTRIAL IRRADIANCE= ,1PE10.4,11H W/M2-SR-MU)
      END
```

```
      FUNCTION SOLARS(WAVL)
C
C     CALCULATE THE INTENSITY OF THE SOLAR SPECTRUM FOR WAVELENGTH (WAVL
C
C     UNITS:
C           SOLARS ... WATTS M-2 MICRON-1
C           WAVL ..... MICRONS
      IF (WAVL.LT.0.15.OR.WAVL.GE.100.) GO TO 100
      IF (WAVL.GE.0.15.AND.WAVL.LE.0.43) GO TO 200
      IF (WAVL.GT.0.43.AND.WAVL.LT.0.58) GO TO 300
      GO TO 400
  100 SOLARS=0.0
      RETURN
  200 Z=((WAVL-.415)/0.68)**2
      SOLARS=1775.*EXP(Z/2)
      RETURN
  300 CONTINUE
      SOLARS=(-61142.*WAVL**4)+(1344477.*WAVL**3)-(110296.*
     1       WAVL**2)
      SOLARS=(SOLARS+39952.7*WAVL-5371.)/100.
      RETURN
  400 IF (WAVL.GE.2.5) GO TO 500
      SOLARS=5331.9*EXP(-1.959*WAVL)
      RETURN
  500 SOLARS=2288.38*(WAVL**(-3.9765))
      RETURN
      END
```

```
      FUNCTION SMOON(WLAM,ANGLE)
C
C     CALCULATE THE INTENSITY OF MOONLIGHT FOR WAVELENGTH (WLAM)
C     AND PHASE ANGLE (ANGLE)
C
C     UNITS:
C           SMOON ... WATTS M-2 MICRON-1
C           ANGLE ... DEGREES
C           WLAM .... MICRONS
      SMOON=0.0
      IF (ANGLE.GT.160.) RETURN
      SMOON=(3.426E-9*ANGLE**4-1.63E-6*ANGLE**3+3.01E-4*
     1      ANGLE**2-.0266*ANGLE+.9882)*100.
      ALBED=0.4
      IF (WLAM.GE.5.) GO TO 200
      IF (WLAM.GT.2.8) GO TO 100
      IF (WLAM.LE.1.) ALBED=3.9633*WLAM**4-10.7306*WLAM**3+
     1   10.2188*WLAM**2-3.8208*WLAM+.5512
      IF (WLAM.GT.1.) ALBED=.0482*WLAM**4-.3283*WLAM**3+
     1   .7584*WLAM**2-.5745*WLAM+.2808
      GO TO 200
  100 ALBED=.350+(.500-.350)*(WLAM-2.8)/2.2
  200 SMOON=2.04472E-07*SOLARS(WLAM)*ALBED*SMOON
      RETURN
      END
```

```
      FUNCTION JPASCT(ICAT)
C     THIS FUNCTION CONVERTS THE INTEGER CODE FOR PASQUILL CATEGORY
C     TO THE ALPHA CHARACTER
      DIMENSION NPASCT(6)
      DATA NPASCT/1HA,1HB,1HC,1HD,1HE,1HF/
       JPASCT=NPASCT(ICAT)
         RETURN
      END
```

```
      SUBROUTINE PFUNC(IDN)
      LOGICAL ISPOT,LOREAD,N16
      DIMENSION PFSPOT(16),PFH(65)
      COMMON /SPOTLO/ISPOT,LOREAD,N16
      COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /CGEOM/COSGM,COSBT,COSIN
      COMMON /BASPOT/ANG(65),SUM(65),WVL(16),NWVL,ALB(16),BS(16),
     +BE(16),SINGWV,PF(65),LMAX
C
C     THIS SUBROUTINE PERFORMS PHASE FUNCTION READING AND INTERPOL-
C     ATION OPERATIONS FOR THE SPOT AND BASCAT MODULES. THE FILE IN
C     WHICH THE PHASE FUNCTION DATA RESIDES IS PFNDAT. USERS MAY IN-
C     SERT PHASE FUNCTIONS OF THEIR OWN SPECIFICATION INTO PFNDAT
C     UNDER AN ID NUMBER 0. THIS PHASE FUNCTION MAY HAVE ANY AR-
C     BITRARY NORMALIZATION, SINCE PFUNC WILL RENORMALIZE IT TO
C     CONFORM TO THE NORMALIZATIONS USED IN SPOT AND BASCAT.
C     FOR FURTHER DETAILS ON THIS PROCEDURE,
C     THE USER IS REFERRED TO CHAPTER 16 OF THE EOSAEL 80 TECHNICAL
C     DOCUMENTATION MANUAL, WHERE THE STRUCTURE, USE, AND MODIFICATION
C     OF THE PFNDAT FILE IS DISCUSSED.
C
C***  NOTE *** FOR USER-DEFINED PHASE FUNCTIONS (IDN=0), THIS ROUTINE
C               WILL INTERPOLATE OVER WAVELENGTH AND ANGLE FOR THE SPOT
C               MODULE. FOR THE BASCAT MODULE, HOWEVER, NO WAVELENGTH
C               INTERPOLATION IS PERFORMED FOR THE USER-DEFINED PHASE
C               FUNCTION. ONLY RENORMALIZATION IS PERFORMED IN THIS
C               LATTER CASE.
C
C***  MAXID IS THE NUMBER OF DIFFERENT DISTRIBUTIONS - (PHASE FUNCTIONS)
C
      MAXID=12
C
C***      CHECK THE ALLOWABLE RANGE OF DISTRIBUTIONS
C
      IF((IDN.GT.MAXID).OR.(IDN.LT.0))GO TO 491
C
C***     DATA INITIALIZATIONS
C
      DO 40 I=1,65
   40 PF(I)=0.
      ALBE=0.
      BEX=0.
      BSC=0.
C
C***            ANGULAR READ BLOCK COMMON TO SPOT AND BASCAT
C
      I=0
C
C**** READ IN ANGLES AT WHICH PHASE FUNCTION IS DEFINED
C
   50 I=I+1
      L1=(I-1)*11+1
      L11=L1+10
      IF(L11.EQ.66)L11=65
      IF(L11.GT.66)GO TO 492
      READ(IPHFUN,60)(ANG(L),L=L1,L11)
   60 FORMAT(11(F6.2,1X))
C
C**** CHECK THIS ROW OF DATA FILE FOR TERMINATION SENTINEL
C
      DO 100 K=L1,L11
      IF((ANG(K).GE.999.99).AND.(L11.LT.65))GO TO 200
      IF(K.EQ.65)GO TO 200
  100 CONTINUE
      GO TO 50
C
C***  LMAX IS THE NBR OF ANGLE AND NBR OF PHASE FUNCTION VALUES+1 HERE.
C
  200 LMAX=K
```

```
      DO 250 L=1,LMAX
  250 ANG(L)=COS(ANG(L)*PIRAD)
C
C***  REDUCE DUE TO SENTINEL OF 999.99
C
      IF(L11.LT.65)LMAX=K-1
      IDNM=IDN-1
      KMAX=IFIX(ALOG(FLOAT(LMAX-1))/ALOG(2.0)+0.1)
C
C***  RESET PARAMETERS FOR BASCAT PROCESSING IF APPROPRIATE.
C
      IF(ISPOT) GO TO 260
      NWVL=1
      WVL(1)=SINGWV
  260 CONTINUE
      IF((IDN.EQ.1).OR.(IDN.EQ.0.)) GO TO 1050
C
C***  READ PAST AEROSOL DATA NOT OF CURRENT INTEREST
C
      DO 1000 I=1,IDNM
      DO 1000 II=1,16
      READ(IPHFUN,300) IANG,ID,WAVE,ALBE,BEX,BSC
  300 FORMAT(2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))
      IF(IANG.NE.LMAX) GO TO 493
      READ(IPHFUN,400) (PF(L),L=1,LMAX)
  400 FORMAT(6(E12.6,1X))
 1000 CONTINUE
 1050 CONTINUE
C
C***  OMIT WAVELENGTH CHECKS FOR USER-DEFINED PHASE FUNCTION.
C
      IF(IDN.EQ.0) GO TO 1070
C
C***  THE NEXT LOOP PERFORMS THE FOLLOWING OPERATIONS :
C        (A) IT VERIFIES WHETHER OR NOT ALL INPUT WAVELENGTHS LIE
C            WITHIN THE 0.2-12.0 MICROMETER RANGE (WITH LIMITS
C            EXTENDED TO PLUS OR MINUS 5%).
C        (B) IF THE PHASE FUNCTION IS NOT USER-SPECIFIED AND THE INPUT
C            WAVELENGTH BAND LIES WITHIN THE 0.2-2.0 BAND, THEN
C            INTERPOLATION IS NOT POSSIBLE DUE TO THE PRESENCE OF
C            ONLY TWO DATA POINTS (AT 0.55 AND 1.06) IN THIS REGION.
C            AS A RESULT, THE 0.55 AND 1.06 VALUES ARE ASSIGNED TO
C            INDIVIDUAL POINTS IN THE INPUT WAVELENGTH BAND. THOSE
C            POINTS WITH WAVELENGTH VALUES LESS THAN OR EQUAL TO 0.8
C            MICROMETERS ARE ASSIGNED THE 0.55 DATA. ALL OTHER WAVE-
C            LENTHS ARE ASSIGNED THE 1.06 DATA.
C        (C) IF AN INPUT WAVELENGTH LIES OUTSIDE OF THE 3-5 OR 8-12
C            MICROMETER BANDS, BUT IS WITHIN 5% OF AN EXTREMUM FOR
C            THESE BANDS, IT IS RESET TO THE EXTREMUM WAVELENGTH
C            VALUE.
C        (D) IF AN INPUT WAVELENGTH LIES BETWEEN BANDS, EXECUTION IS
C            TERMINATED AND AN ERROR MESSAGE IS PRINTED.
C
      DO 1060 I=1,NWVL
      IF((WVL(I).LT.0.19).OR.(WVL(I).GT.12.049)) GO TO 504
      IF(WVL(I).LE.0.8) WVL(I)=0.55
      IF((WVL(I).GT.0.8).AND.(WVL(I).LE.2.1)) WVL(I)=1.06
      IF(WVL(I).GT.12.0) WVL(I)=12.0
      IF((WVL(I).GT.5.0).AND.(WVL(I).LE.5.25)) WVL(I)=5.0
      IF((WVL(I).GE.7.6).AND.(WVL(I).LT.8.0)) WVL(I)=8.0
      IF((WVL(I).GE.2.85).AND.(WVL(I).LT.3.0)) WVL(I)=3.0
      IF((WVL(I).GT.2.1).AND.(WVL(I).LT.2.85)) GO TO 498
      IF((WVL(I).GT.5.25).AND.(WVL(I).LT.7.6)) GO TO 498
 1060 CONTINUE
 1070 CONTINUE
C
C***  MAIN INTERPOLATION LOOP
C
      DO 2000 I=1,NWVL
      IF(I.GT.1) GO TO 1260
```

```
 1100 CONTINUE
      READ(IPHFUN,300) IANG,ID,WAVEH,ALBH,BEXH,BSCH
      READ(IPHFUN,400) (PFH(L),L=1,LMAX)
      IF((IDN.EQ.0).AND.(.NOT.ISPOT)) GO TO 1280
      IF(WVL(I).LT.WAVEH) GO TO 1100
 1150 CONTINUE
      READ(IPHFUN,300) IANG,ID,WAVE,ALBE,BEX,BSC
      READ(IPHFUN,400) (PF(L),L=1,LMAX)
      IF(WVL(I).LE.WAVE) GO TO 1260
 1160 CONTINUE
      WAVEH=WAVE
      ALBH=ALBE
      BEXH=BEX
      DO 1240 L=1,LMAX
 1240 PFH(L)=PF(L)
      GO TO 1150
 1260 CONTINUE
C
C***  GO TO NEXT WAVELENGTH INTERPOLATION INTERVAL IF INPUT WAVE-
C***  LENGTH IS GREATER THAN THE MAXIMUM OF THE CURRENT ONE.
C
      IF(WVL(I).GT.WAVE) GO TO 1160
C
C***  RENORMALIZE LOWER END OF INTERPOLATION INTERVAL
C
 1280 CONTINUE
      SUM(1)=0.0
      DO 1200 L=2,LMAX
 1200 SUM(L)=((ANG(L-1)-ANG(L))*(PFH(L-1)+PFH(L))/4.0)+SUM(L-1)
      SUMT=SUM(LMAX)
      DO 1250 L=1,LMAX
 1250 PFH(L)=PFH(L)/SUMT
C
C***  BRANCH TO FINAL PROCEDURE FOR BASCAT USER-DEFINED PHASE
C***  FUNCTION IF APPROPRIATE.
C
      IF((IDN.EQ.0).AND.(.NOT.ISPOT)) GO TO 2500
C
C***  RENORMALIZE UPPER END OF INTERPOLATION INTERVAL.
C
      SUM(1)=0.0
      DO 1400 L=2,LMAX
 1400 SUM(L)=((ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0)+SUM(L-1)
      SUMT=SUM(LMAX)
      DO 1450 L=1,LMAX
 1450 PF(L)=PF(L)/SUMT
C
C***  BRANCH TO BASCAT WAVELENGTH INTERPOLATION PROCEDURE IF
C***  APPROPRIATE.
C
      IF(.NOT.ISPOT) GO TO 2500
C
C***  PERFORM HALVING SEARCH FOR COSINES IN PHASE FUNCTION DATA FILE
C***  WHICH BRACKET COSINE INPUT FROM SPOT.
C
      L=1
      LL=LMAX-1
      DO 1300 K=1,KMAX
      LL=LL/2
      L=L+LL
      AT=COSIN-ANG(L)
      IF(AT.GT.0.) L=L-LL
 1300 CONTINUE
C
C***  PERFORM SPOT EXTINCTION COEFFICIENT AND PHASE FUNCTION
C***  INTERPOLATIONS OVER WAVELENGTH AND ANGLE.
C
      FACANG=(COSIN-ANG(L))/(ANG(L+1)-ANG(L))
      FACWVL=(WVL(I)-WAVEH)/(WAVE-WAVEH)
      BE(I)=BEXH+(BEX-BEXH)*FACWVL
```

```
      PFSPOT(I)=PFH(L)*(1.-FACANG-FACWVL+FACANG*FACWVL)+
     +PFH(L+1)*(FACANG-FACANG*FACWVL)+PF(L)*(FACWVL-FACANG*FACWVL)+
     +PF(L+1)*(FACANG*FACWVL)
 2000 CONTINUE
C
C***  LOAD FIRST NWVL ANGLES OF OUTPUT ARRAY PF( ) WITH INTERPOLATED
C***  RESULTS FOR SPOT.
C
      DO 2200 N=1,NWVL
 2200 PF(N)=PFSPOT(N)/(4.*PI)
C
C***  FINAL EXIT FOR SPOT PROCESSING.
C
      GO TO 500
 2500 CONTINUE
      FACWVL=0.
      IF(IDN.EQ.0) GO TO 2700
C
C***  BASCAT ALBEDO, EXTINCTION COEFFICIENT, AND PHASE FUNCTION
C***  INTERPOLATION OVER WAVELENGTH.
C
      FACWVL=(WVL(1)-WAVEH)/(WAVE-WAVEH)
 2700 CONTINUE
      DO 2800 L=1,LMAX
 2800 PF(L)=PFH(L)+(PF(L)-PFH(L))*FACWVL
      ALB(1)=ALBH+(ALBE-ALBH)*FACWVL
      BE(1)=BEXH+(BEX-BEXH)*FACWVL
C
C***  FINAL EXIT FOR BASCAT USER-DEFINED PHASE FUNCTION PROCEDURE.
C
      IF(IDN.EQ.0) GO TO 500
C
C***  FINAL BASCAT PHASE FUNCTION RENORMALIZATION.
C
      SUM(1)=0.0
      DO 2900 L=2,LMAX
 2900 SUM(L)=((ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0)+SUM(L-1)
      SUMT=SUM(LMAX)
      DO 2950 L=1,LMAX
 2950 PF(L)=PF(L)/SUMT
C
C***  FINAL EXIT FOR BASCAT PROCESSING.
C
      GO TO 500
C
C***            ERROR EXIT BLOCK COMMON TO SPOT AND BASCAT
C
  491 CONTINUE
      WRITE(IOOUT,495)
  495 FORMAT(1H0,20X,58H***PFUNC ERROR*** AEROSOL ID NUMBER OUT OF ALLOW
     +ABLE RANGE //)
      STOP
  492 CONTINUE
      WRITE(IOOUT,496)
  496 FORMAT(1H0,20X,83H***PFUNC ERROR*** READ TERMINATION SENTINEL NOT
     +FOUND OR NUMBER OF ANGLES EXCEED 65 //)
      STOP
  493 CONTINUE
      WRITE(IOOUT,497)
  497 FORMAT(1H0,20X,93H***PFUNC ERROR*** NUMBER OF SPECIFIED ANGLES AND
     + NUMBER OF PHASE FUNCTION VALUES DO NOT MATCH //)
      STOP
  498 CONTINUE
      WRITE(IOOUT,499)
  499 FORMAT(1H0,10X, 79H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WV
     +L INPUT ARRAY DO NOT LIE WITHIN /1H ,45HWAVELENGTH BANDS COVERED B
     +Y PFNDAT DATA BASE /)
      STOP
  504 CONTINUE
      WRITE(IOOUT,505)
```

```
  505 FORMAT(1H0,10X,109H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WVPFU02820
     +L ARRAY DO NOT LIE WITHIN OVERALL ACCEPTABLE RANGE OF 0.2-12.0 /) PFU02830
      STOP                                                              PFU02840
 500       RETURN                                                       PFU02850
      END                                                               PFU02860
```

```
      SUBROUTINE XSCALE(WAVE,VIS,EXT55,XSTRN,IERR,ISLT,IFOG,RANGE,ANGLE)XSC00010
C    THE PURPOSE OF THIS ROUTINE IS TO A) FIND THE HORIZONTAL EXTINCTIONXSC00020
C      IN FOG AT THE WAVELENGTHS SPECIFIED BELOW FROM THE EXTINCTION    XSC00030
C      AT .55 UM OR B) THE EXTINCTION ALONG A SLANT PATH AT ALLOWED     XSC00040
C      WAVELENGTHS FROM THE EXTINCTION AT .55 UM: FOG TYPE 1, 2, OR 3,  XSC00050
C      MUST BE SPECIFIED FOR SLANT PATHS.                               XSC00060
C  *** VISIBILITY = 88. OR 89. IS NOT ALLOWED AS THIS IS USED AS AN     XSC00070
C      INDICATOR THAT XSCALE IS BEING CALLED AS A SUBROUTINE FROM EITHER XSC00080
C      SPOT(88.), LOWTRAN(88.), OR CWIC(89.), NOT EOMAIN!               XSC00090
C      WAVE=LAMDA IN UM - MUST BE .55, 1.06, 3.0-5.0, 8.0-12.05.        XSC00100
C      ***  ALL EXTN'S ARE IN KM**-1                                    XSC00110
C      EXT55 = EXTINCTION AT .55 UM                                     XSC00120
C      EXT106 = EXTINCTION AT 1.06 UM                                   XSC00130
C      EXT35 = EXTINCTION FROM 3.0 TO 5.0 UM                            XSC00140
C      EXT812 = EXTINCTION FROM 8.0 TO 12.0 UM                          XSC00150
C      VIS= VISIBILITY IN KM -OR- EXT55 IN KM**-1                       XSC00160
C      EXT55 IS ** NOT ** CHANGED BY THIS ROUTINE.                      XSC00170
C*****************************************************************      XSC00180
C      INPUT:  THERE IS A MAXIMUM OF 3 CARDS TO EXECUTE THIS MODULE     XSC00190
C      THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OF     XSC00200
C      THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN.        XSC00210
C      THE CARDS ARE INPUT WITH FORMAT (A4,6X,5(F6.2,1X)).              XSC00220
C      EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4         XSC00230
C      FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 6 COL PER           XSC00240
C      FIELD BEGINNING IN COL 11, WITH A BLANK BETWEEN EACH SUBSEQUENT  XSC00250
C      FIELD.  THE CARDS ARE NOT ORDER DEPENDENT.                       XSC00260
C      IF GEOMET OPTION IS BEING USED, THEN ONLY THE IDENTIFIER HORZ,
C      SLNH, OR SLNS IS TO BE READ IN (NO ADDITIONAL PARAMETERS NEEDED).
C      FOG         FOG TYPE, RAIN RATE (MM/HR): RAIN RATE ONLY NEEDED   XSC00270
C                  WHEN FOG TYPE=4.                                     XSC00280
C      HORZ        HORDIS (KM)             :HORIZONTAL PATH CALCULATION XSC00300
C      SLNH        HORDIS (KM), ANGLE (DEGREES): SLANT PATH CALCULATION XSC00310
C      SLNS        SLTDIS (KM), ANGLE (DEGREES): SLANT PATH CALCULATION XSC00320
C      PLOT        WRITE SLANT PATH EXTINCTION, AT INPUT WAVELENGTH, AND
C                  ALTITUDE TO NPLOTU (SEE COMMON BLOCK IOUNIT): THE
C                  FIRST RECORD WILL BE THE NUMBER OF POINTS TO BE WRITTEN.
C                  FORMATS: RECORD 1 (I5), SUBSEQUENT RECORDS (2(E10.4,1X))
C      GO         SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FOR       XSC00330
C                 THIS CALL.  NOTE THAT IF A DATA CARD IS NOT READ      XSC00340
C                 THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLS       XSC00350
C                 TO THE MODULE ARE STILL IN EFFECT.                    XSC00360
C      ALL THE FOLLOWING FOG TYPES ARE RELEVANT TO HORIZONTAL PATHS,
C      BUT ONLY FOG TYPES 1, 2, OR 3 ARE ALLOWED FOR SLANT PATH CALCULATIONS
C      FOG TYPE=1. FOR MARITIME ARTIC                                   XSC00380
C              =2. FOR MARITIME POLAR                                   XSC00390
C              =3. FOR CONTINENTAL POLAR                                XSC00400
C              =4. FOR RAIN                                             XSC00410
C              =5. FOR SNOW                                             XSC00420
C      HORDIS - HORZONTAL DISTANCE IN KM.                               XSC00430
C      SLTDIS - SLANT PATH DISTANCE IN KM.                              XSC00440
C      ANG= LOOK ANGLE FROM HORIZONTAL IN DEGREES                       XSC00450
C   N.B.    ONE OF THE FOLLOWING COMBINATIONS MUST BE SUPPLIED          XSC00460
C             FOR SLANT PATH CALCULATIONS.                              XSC00470
C      HORDIS AND ANG ** OR ** SLTDIS AND ANG                           XSC00480
C*****************************************************************      XSC00490
C   OUTPUT                                                              XSC00500
C    TRANSMISSION AT APPROPRIATE WAVELENGTH FOR SLANT OR HORIZONTAL PATHXSC00510
C                                                                       XSC00520
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     XSC00530
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUXSC00540
      COMMON /GEOMET/PTS(15),IGEOSW                                     XSC00550
      DIMENSION TYPE(6),DAT(6)                                          XSC00560
      LOGICAL NOLO                                                      XSC00570
      DATA TYPE /4HFOG ,4HHORZ,4HSLNH,4HSLNS,4HPLOT,4HGO  /             XSC00580
      DATA A0,A1,A2,NPLT/0.1425,0.1475,-0.0017,0/                       XSC00590
C     USE VIS=88., OR 89. AS AN INDICATOR THAT XSCALE HAS BEEN CALLED   XSC00600
C     AS A SUBROUTINE FROM OTHER PROGRAMS - NOT EOMAIN!                 XSC00610
      IF (VIS.LT.87.9.OR.VIS.GT.89.1) GO TO 8                           XSC00620
      ANG=ANGLE                                                         XSC00630
C     FIND ELEVATION ANGLE FROM ZENITH ANGLE IN SPOT AND LOWTRAN        XSC00640
```

```
      IF (VIS.GT.87.9.AND.VIS.LT.88.1) ANG=90.-ANGLE                    XSC00650
      IF (VIS.GT.87.9.AND.VIS.LT.88.1.AND.ANGLE.GT.90.) ANG=ANGLE-90.   XSC00660
      HORDIS=0.                                                         XSC00670
      SLTDIS=0.                                                         XSC00680
      NOLO=.FALSE.                                                      XSC00690
C     ISLT=0=HORIZONTAL; ISLT GT 0 = SLANT.                             XSC00700
      IF (ISLT.EQ.0) HORDIS=RANGE                                       XSC00710
      IF (ISLT.EQ.0) ISLANT=0                                           XSC00720
      IF (ISLT.GT.0) SLTDIS=RANGE                                       XSC00730
      IF (ISLT.GT.0) ISLANT=1                                           XSC00740
      GO TO 6                                                           XSC00750
8     CONTINUE                                                          XSC00760
      NOLO=.TRUE.                                                       XSC00770
      DO 9 I=1,3                                                        XSC00780
      READ (IOIN,500) (DAT(J),J=1,5)                                    XSC00790
      IF (DAT(1).EQ.TYPE(1)) GO TO 1                                    XSC00800
      IF (DAT(1).EQ.TYPE(2)) GO TO 2                                    XSC00810
      IF (DAT(1).EQ.TYPE(3)) GO TO 3                                    XSC00820
      IF (DAT(1).EQ.TYPE(4)) GO TO 4                                    XSC00830
      IF (DAT(1).EQ.TYPE(5)) GO TO 5
      IF (DAT(1).EQ.TYPE(6)) GO TO 6                                    XSC00840
C     ERROR CHECK                                                       XSC00850
      GO TO 7                                                           XSC00860
C     ADVERSE WEATHER INDICATOR AND OPTIONAL RAIN RATE:                 XSC00870
1     IFOG=IFIX(DAT(2))                                                 XSC00880
      RNRT=DAT(3)                                                       XSC00890
      GO TO 9                                                           XSC00900
C     HORIZONTAL DISTANCE FOR HORIZONTAL PATH CALC.                     XSC00910
2     HORDIS=DAT(2)                                                     XSC00920
      ISLANT=0                                                          XSC00930
      GO TO 9                                                           XSC00940
C     HORIZONTAL DISTANCE AND ANGLE FOR SLANT PATH CALC.                XSC00950
3     HORDIS=DAT(2)                                                     XSC00960
      ANG=DAT(3)                                                        XSC00970
      ISLANT=1                                                          XSC00980
      GO TO 9                                                           XSC00990
C     SLANT DISTANCE AND ANGLE FOR SLANT PATH CALC.                     XSC01000
4     SLTDIS=DAT(2)                                                     XSC01010
      ANG=DAT(3)                                                        XSC01020
      ISLANT=1                                                          XSC01030
      GO TO 9
C     SET PLOT FLAG
5     NPLT=1
9     CONTINUE                                                          XSC01040
6     CONTINUE                                                          XSC01050
      IF (NOLO) WRITE (IOOUT,600)                                       XSC01060
      IF(IGEOSW.NE.1)GO TO 88                                           XSC01070
      HORDIS=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2)                XSC01080
      SLTDIS=SQRT(HORDIS**2+(PTS(3)-PTS(6))**2)                         XSC01090
      ANG=ACOS(HORDIS/SLTDIS)/PIRAD                                     XSC01100
   88 CONTINUE                                                          XSC01110
C     WAVELENGTH ERROR CHECK                                            XSC01120
      IF ((WAVE.GT..4.AND.WAVE.LE.2.).OR.(WAVE.GE.3..AND.WAVE.LE.5.)    XSC01130
     1 .OR.(WAVE.GE.8.AND.WAVE.LE.12.05)) GO TO 10                      XSC01140
      WRITE (IOOUT,1600) WAVE                                           XSC01150
      IERR=1                                                            XSC01160
      XSTRN=1.                                                          XSC01170
      RETURN                                                            XSC01180
10    CONTINUE                                                          XSC01190
      IF (NOLO.AND.IFOG.EQ.1) WRITE (IOOUT,800)                         XSC01210
      IF (NOLO.AND.IFOG.EQ.2) WRITE (IOOUT,900)                         XSC01220
      IF (NOLO.AND.IFOG.EQ.3) WRITE (IOOUT,950)                         XSC01230
      IF (NOLO.AND.IFOG.EQ.4) WRITE (IOOUT,1000)                        XSC01240
      IF (NOLO.AND.IFOG.EQ.5) WRITE (IOOUT,1100)                        XSC01250
      IF (ISLANT.GT.0) GO TO 11                                         XSC01200
      IF (NOLO) WRITE (IOOUT,1200)                                      XSC01260
11    IF (NOLO.AND.ISLANT.GE.1) WRITE (IOOUT,1400) WAVE                 XSC01270
      IF (ISLANT.GE.1.AND.(IFOG.LE.0.OR.IFOG.GE.4)) WRITE (IOOUT,2100)
      IF (ISLANT.GE.1.AND.(IFOG.LE.0.OR.IFOG.GE.4)) IFOG=1
      EXTN=EXT55
```

```
      IF (ISLANT.GE.1.AND.WAVE.GE..8) GO TO 101
12     IF (ISLANT.GE.1) CALL SLANT(EXTN,HORDIS,SLTDIS,ANG,AVEX55,IERR,  XSC01280
     1 WAVE,NPLT)                                                        XSC01290
      IF (IERR.EQ.1) XSTRN =1.                                           XSC01300
      IF (IERR.EQ.1) RETURN                                              XSC01310
      EXTN=EXT55                                                         XSC01320
      IF (ISLANT.GE.1) EXTN=AVEX55                                       XSC01330
      XSPATH=HORDIS                                                      XSC01340
      IF (ISLANT.GE.1) XSPATH=SLTDIS                                     XSC01350
      IF (ISLANT.EQ.0) GO TO 100                                         XSC01360
C     SLANT PATH EXTINCTION                                              XSC01370
      XSTRN =EXP(-XSPATH*EXTN)                                           XSC01380
      IF (NOLO) WRITE (IOOUT,1500) WAVE,EXTN,XSPATH,XSTRN,ANG            XSC01390
      RETURN                                                             XSC01400
100   IF (IFOG.NE.4) GO TO 101                                           XSC01410
C     RAIN - ALL WAVELENGTHS                                             XSC01420
      IF (NOLO.AND.RNRT.LE.0.0) WRITE (IOOUT,550)                        XSC01430
      IF (RNRT.LE.0.0) RNRT=1.                                           XSC01440
      RNEXTN=A0+A1*RNRT+A2*RNRT**2                                       XSC01450
      XSTRN =EXP(-XSPATH*RNEXTN)                                         XSC01460
      IF (NOLO) WRITE (IOOUT,1550) RNRT,RNEXTN,XSPATH,XSTRN              XSC01470
      RETURN                                                             XSC01480
  101 IF (ABS(WAVE-1.06).LT..01) GO TO 400                               XSC01490
      IF (WAVE.GE.3..AND.WAVE.LE.5.) GO TO 200                           XSC01500
      IF (WAVE.GE.8..AND.WAVE.LE.12.) GO TO 300                          XSC01510
      XSTRN=EXP(-XSPATH*EXTN)
      RETURN                                                             XSC01520
  200 CONTINUE                                                           XSC01530
C     3.0 TO 5.0 RANGE                                                   XSC01540
C     MA                                                                 XSC01550
      IF (IFOG.EQ.1) EXT35=10.**(0.0345+1.03*ALOG10(EXTN))               XSC01560
C     MP                                                                 XSC01570
      IF (IFOG.EQ.2) EXT35=10.**(-0.38+1.32*ALOG10(EXTN))                XSC01580
C     CP                                                                 XSC01590
      IF (IFOG.EQ.3) EXT35=10.**(-0.82+1.58*ALOG10(EXTN))                XSC01600
      IF(IFOG.NE.5) EXTN=EXT35
      IF (ISLANT.GE.1) GO TO 12
C     SNOW                                                               XSC01610
      IF (IFOG.EQ.5) EXT35=10.0**(1.05*ALOG10(EXTN)+.021)                XSC01620
      XSTRN =EXP(-XSPATH*EXT35)                                          XSC01630
      IF (NOLO) WRITE (IOOUT,1700) EXT35,XSPATH,XSTRN                    XSC01640
      RETURN                                                             XSC01650
  300 CONTINUE                                                           XSC01660
C     8.0 TO 12.0 RANGE                                                  XSC01670
C     MA                                                                 XSC01680
      IF (IFOG.EQ.1) EXT812=10.**(-.45+1.19*ALOG10(EXTN))                XSC01690
C     MP                                                                 XSC01700
      IF (IFOG.EQ.2) EXT812=10.**(-1.01+1.51*ALOG10(EXTN))               XSC01710
C     CP                                                                 XSC01720
      IF (IFOG.EQ.3) EXT812=10.**(-1.65+1.82*ALOG10(EXTN))               XSC01730
      IF(IFOG.NE.5) EXTN=EXT812
      IF (ISLANT.GE.1) GO TO 12
C     SNOW                                                               XSC01740
      IF (IFOG.EQ.5) EXT812=10.0**(.993*ALOG10(EXTN)+.114)               XSC01750
      XSTRN =EXP(-XSPATH*EXT812)                                         XSC01760
      IF (NOLO) WRITE (IOOUT,1800) EXT812,XSPATH,XSTRN                   XSC01770
      RETURN                                                             XSC01780
  400 CONTINUE                                                           XSC01790
C     1.06 RANGE                                                         XSC01800
C     MA, MP, AND CP                                                     XSC01810
      EXT106=AMIN1(10.**(-0.14+1.16*ALOG10(EXTN)),EXTN)
      EXTN=EXT106
      IF (ISLANT.GE.1) GO TO 12
C     SNOW - ASSUME THAT THE EXTINCTION AT 1.06 IS THE SAME AS AT .55 UMXSC01870
      IF (IFOG.EQ.5) EXT106=EXT55                                        XSC01880
      XSTRN =EXP(-XSPATH*EXT106)                                         XSC01890
      IF (NOLO) WRITE (IOOUT,1900) EXT106,XSPATH,XSTRN                   XSC01900
      RETURN                                                             XSC01910
7     WRITE (IOOUT,2000) (DAT(J),J=1,4)                                  XSC01920
      XSTRN=1.                                                           XSC01930
```

```
      IERR=1                                                            XSC01940
      RETURN                                                            XSC01950
C                                                                       XSC01960
500   FORMAT (A4,6X,5(F6.2,1X))                                         XSC01970
550   FORMAT(1H0,50HXSCALE WARNING - RAIN RATE IS .LE. 0.0, RAIN RATE , XSC01980
     +20HSET EQUAL TO 1 MM/HR,/)                                        XSC01990
  600 FORMAT (1H ,///,1X,50X,14HOPTIONS CHOSEN)                         XSC02000
  800 FORMAT (1H ,50X,14HMARITIME ARTIC)                                XSC02010
  900 FORMAT (1H ,50X,14HMARITIME POLAR)                                XSC02020
  950 FORMAT (1H ,50X,17HCONTINENTAL POLAR)                             XSC02030
 1000 FORMAT (1H ,50X,4HRAIN)                                           XSC02040
 1100 FORMAT (1H ,50X,4HSNOW)                                           XSC02050
 1200 FORMAT (1H ,50X,15HHORIZONTAL PATH/)                              XSC02060
 1400 FORMAT (1H ,50X,15HSLANT PATH FOR ,F8.3,8H MICRONS/)              XSC02070
 1500 FORMAT (1X,23X,20HSLANT EXTINCTION AT ,F5.2,8H MICRONS,           XSC02080
     1      4X,11HDISTANCE    ,12HTRANSMISSION,6X,5HANGLE/,1X,40X,      XSC02090
     2          26HKM**-1                    KM,/                       XSC02100
     3          ,1X,38X,F8.3,12X,F8.3,5X,E9.3,7X,F7.2)                  XSC02110
 1550 FORMAT (1H ,20X,27HEXTINCTION FOR RAIN RATE OF,F5.2,6H MM/HR,     XSC02120
     1      5X,11HDISTANCE    ,12HTRANSMISSION/,1X,40X,                 XSC02130
     2          26HKM**-1                    KM,/                       XSC02140
     3          ,1X,38X,F8.3,12X,F8.3,5X,E9.3)                          XSC02150
 1600 FORMAT (1H ,18H***** WAVELENGTH (,F9.3,10H) OUTSIDE ,             XSC02160
     1          10HALLOWABLE ,29HRANGE (1.06,3.0-5.0,8.0-12.0           XSC02170
     2          31HMICRONS) - CONTROL RETURNED TO ,                     XSC02180
     3          17HMAIN FROM XSCALE,)                                   XSC02190
 1700 FORMAT (1H ,25X,37HEXTINCTION FROM 3.0 TO 5.0 MICRONS             XSC02200
     1          11HDISTANCE    ,12HTRANSMISSION/,1X,40X,                XSC02210
     2          26HKM**-1                    KM,/                       XSC02220
     3          ,1X,38X,F8.3,13X,F8.3,5X,E9.3)                          XSC02230
 1800 FORMAT (1H ,25X,37HEXTINCTION FROM 8.0 TO 12.0 MICRONS            XSC02240
     1          12H DISTANCE    ,12HTRANSMISSION/,1X,40X,               XSC02250
     2          26HKM**-1                    KM,/                       XSC02260
     3          ,1X,38X,F8.3,12X,F8.3,5X,E9.3)                          XSC02270
 1900 FORMAT (1H ,25X,37HEXTINCTION AT 1.06 MICRONS                     XSC02280
     1          11HDISTANCE    ,12HTRANSMISSION/,1X,40X,                XSC02290
     2          26HKM**-1                    KM,/                       XSC02300
     3          ,1X,38X,F8.3,12X,F8.3,5X,E9.3)                          XSC02310
2000  FORMAT (1H ,44HUNKNOWN CARD TYPE.  CONTROL RETURNED TO MAIN,      XSC02320
     +13H FROM XSCALE,,/,1X,A4,6X,5(F6.2,1X))                           XSC02330
2100  FORMAT(1H ,40HINCORRECT FOG TYPE FOR SUBROUTINE SLANT,,/,1X,
     1 21HFOG TYPE CHANGED TO 1/)
      END                                                               XSC02340
```

```
      SUBROUTINE SLANT(EXT55,HORDIS,SLTDIS,ANG,AVEX55,IERR,WAVE,NPLT)    SLNT0010
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSLNT0020
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     SLNT0030
C     ALL QUANTITIES IN THIS ROUTINE ARE FOR .55 UM                     SLNT0040
      REAL KMTOM                                                        SLNT0050
C     KILOMETERS TO METERS                                              SLNT0060
      KMTOM=1000.                                                       SLNT0070
      IERR=0                                                            SLNT0080
      TOL=.0001                                                         SLNT0090
  100 IF (HORDIS.LT.TOL.OR.ANG.LT.TOL) GO TO 200                        SLNT0100
C     HORZONTAL DISTANCE AND ANGLE INPUT                                SLNT0110
      VERDIS=HORDIS*TAN(ANG*PIRAD)                                      SLNT0120
      SLTDIS=SQRT(HORDIS**2+VERDIS**2)                                  SLNT0130
      GO TO 300                                                         SLNT0140
200   IF (SLTDIS.LT.TOL.OR.ANG.LT.TOL) GO TO 500                        SLNT0150
C     SLANT DISTANCE AND ANGLE INPUT                                    SLNT0160
      VERDIS=SLTDIS*SIN(ANG*PIRAD)                                      SLNT0170
      HORDIS=SQRT(SLTDIS**2-VERDIS**2)                                  SLNT0180
C     CONVERT TO 20 METER INCREMENTS                                    SLNT0190
300   VERDIS=FLOAT(IFIX((VERDIS+TOL)*KMTOM/20.))*20./KMTOM              SLNT0200
C     LIMIT ON VERTICAL HEIGHT IS 500 METERS                            SLNT0210
      IF (VERDIS*KMTOM.GT.500.) VERDIS=.5                               SLNT0220
      SD=SQRT(HORDIS**2+VERDIS**2)                                      SLNT0230
      IF(SD/SLTDIS.GT.1.01.OR.SD/SLTDIS.LT..99)WRITE(IOOUT,700)SLTDIS,SDSLNT0240
      SLTDIS=SD                                                         SLNT0250
C     FIND NBR OF 20 METER INCREMENTS                                   SLNT0260
      ITOP=IFIX((VERDIS+TOL)*KMTOM/20.)                                 SLNT0270
      IF(ITOP.LT.1) ITOP=1
      VERDIS=FLOAT(ITOP)*20./KMTOM
      EXTN=EXT55                                                        SLNT0280
C     BEGIN TRAPEZODIAL INTEGRATION FOR TAU ( OPTICAL DEPTH)            SLNT0290
      TAU=EXTN/2.                                                       SLNT0300
      ALT=0.0
      NPTS=ITOP+1
      IF (NPLT.EQ.1) WRITE (NPLOTU,352) NPTS
352   FORMAT (I5)
      IF (NPLT.EQ.1) WRITE (NPLOTU,351) EXTN,ALT,WAVE
      DO 400 I=1,ITOP                                                   SLNT0310
C     THESE FORMULAS ARE GOOD ONLY IN 20M INCREMENTS                    SLNT0320
      IF (EXTN.GE.7.0.AND.WAVE.LT.2.0) EXTN=                            SLNT0330
     1 10.**(0.55+0.72*ALOG10(EXTN))                                    SLNT0340
      IF (EXTN.LT.7.0.AND.WAVE.LT.2.0) EXTN=                            SLNT0350
     1 10.**(0.1+1.25*ALOG10(EXTN))                                     SLNT0360
      IF (EXTN.GE.3.3.AND.(WAVE.GE.3.0.AND.WAVE.LT.5.0)) EXTN=          SLNT0370
     1 10.**(0.55+.72*ALOG10(EXTN))                                     SLNT0380
      IF (EXTN.LT.3.3.AND.(WAVE.GE.3.0.AND.WAVE.LT.5.0)) EXTN=          SLNT0390
     1 10.**(0.3+1.2*ALOG10(EXTN))                                      SLNT0400
      IF (EXTN.GE.1.7.AND.(WAVE.GE.8.0.AND.WAVE.LT.12.0)) EXTN=         SLNT0410
     1 10.**(0.5+0.75*ALOG10(EXTN))                                     SLNT0420
      IF (EXTN.LT.1.7.AND.(WAVE.GE.8.0.AND.WAVE.LT.12.0)) EXTN=         SLNT0430
     1 10.**(0.4+1.2*ALOG10(EXTN))                                      SLNT0440
      ALT=FLOAT(I)*20.
      IF (NPLT.EQ.1) WRITE (NPLOTU,351) EXTN,ALT
351   FORMAT (3(E10.4,1X))
  400 TAU=TAU+EXTN                                                      SLNT0450
C     FINISH TRAP INTEGRATION                                           SLNT0460
      TAU=(TAU-EXTN/2.)*.02                                             SLNT0470
C     FIND AVERAGE EXTINCTION VALUE FOR SLANT PATH.                     SLNT0480
      AVEX55=TAU/VERDIS                                                 SLNT0490
      RETURN                                                            SLNT0500
  500 WRITE (IOOUT,600)                                                 SLNT0510
      IERR=1                                                            SLNT0520
      RETURN                                                            SLNT0530
C                                                                       SLNT0540
  600 FORMAT (1X,38HERROR - IMPROPER INPUT FOR SUBROUTINE               SLNT0550
     1             34HSLANT; TRANSMISSION SET EQUAL TO 1)               SLNT0560
700   FORMAT (1H ,18HWARNING FROM SLANT,/,1X,22HTHE VERTICAL DISTANCE , SLNT0570
     +38HEXCEEDS THE 500 METER UPPER LIMIT, OR ,/,1X,10HIS NOT AN ,     SLNT0580
     +29HINTEGER MULTIPLE OF 20 METERS,/,1X,28HSLANT DISTANCE CHANGED FRSLNT0590
     +OM ,F7.4,4H TO ,F7.4,3H KM/)                                      SLNT0600
```

END

SLNT0610

```
      SUBROUTINE TURB(WAVE,IERR)
C    CALCULATES TURBULENCE INDUCED POINTING JITTER AND POWER SPECTRUM
C      FOR LASER TARGET DESIGNATOR AND TERMINAL HOMING SEEKER
C
C***********************************************************************
C         CALCULATION FOR THE DESIGNATOR PATH ARE PERFORMED EACH
C         TIME THIS ROUTINE IS REFERENCED.  THE CALCULATIONS FOR THE
C         SEEKER PATHS ARE PERFORMED ONLY WHEN THE DATA CARDS DVRV,
C         CN2, OR V2 ARE INCLUDED IN THE INPUT SET.
C         THE INPUT IS CARD ORDER INDEPENDENT, WITH THE SINGLE
C         RESTRICTION THAT THE 'GO' CARD MUST BE THE LAST CARD
C         OF THE DATA SET.
C         THE DATA IS COMPLETELY IDENTIFIED BY THE ID IN COLUMNS
C         1-4 OF EACH CARD, FOLLOWED BY UP TO 7(REAL) FIELDS AS
C         NEEDED, WITH 10 COLUMNS PER FIELD BEGINNING IN COL. 11.
C         COMMENTS BELOW.
C
C         THE INPUT FORMAT IS A4,6X,7(E10.4)
C
C         THE INPUT OF A CN1, CN2, V1 AND V2 TYPE DATA CARD IS
C         TERMINATED WHEN THE FIRST COEFFICIENT OF VALUE ZERO
C         IS ENCOUNTERED.  THE REMAINING DATA COEFFICIENTS ON
C         THE CARD, IF ANY, ARE IGNORED.
C
C***********************************************************************
C
C THE FOLLOWING ARE REQUIRED RECORDS FOR AT THE FIRST INPUT SET.
C
C IDENT.   VARS.            DESCRIPTION
C ------   --------     -------------------------------------------
C
C PARM
C            DIAM    LASER TARGET DESIGNATOR APERTURE DIAMETER IN METERS
C            THET    LASER BEAMSPREAD ANGLE IN RADIANS
C            TDOT    LASER BEAM SLUE RATE IN RADIANS/SECOND
C            RANG    DISTANCE FROM DESIGNATOR TO TARGET IN METERS
C            TIME    DURATION OF CALCULATIONS IN SECONDS
C            M       NO. OF FREQUENCIES FOR WHICH POINTING JITTER POWER
C                    SPECTRUM IS TO BE CALCULATED, IF M=0, THEN DEFAULT
C                    TO M=512.
C    ***RANG AND R1V ARE RECALCULATED IF IGEOSW=1
C
C    ***N1 AND N2 ARE CALCULATED WITHIN THE ROUTINE.
C
C            N1      NO OF SEGMENTS IN DESIGNATOR PATH
C            N2      NO. OF SEGMENTS IN SEEKER TO TARGET PATH
C
C         N1 AND N2 ARE SET EQUAL TO THE INDEX OF THE LAST NON-ZERO
C         COEFFICIENT READ INTO CN1 AND CN2 RESPECTIVELY.
C
C CN1
C           IR2      STARTING INDEX VALUE OF CN1(I)
C           (CN1(I), I=IR2,IR2+5)
C            CN1(I)    VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT
C                      WITH ONE VALUE FOR EACH SEGMENT OF RANGE FROM LASER
C                      DESIGNATOR TO TARGET (METERS**(-2/3))
C
C V1
C           IR2      STARTING INDEX VALUE OF V1(I)
C           (V1(I), I=IR2,IR2+5)
C            V1(I)     SET OF VALUES OF CROSSWIND VELOCITY
C                      CORRESPONDING TO EACH SEGMENT OF RANGE FROM LASER
C                      DESIGNATOR TO TARGET (M/SEC)
C
C  DVRV
C             D1V      DIAMETER OF SEEKER APERTURE IN METERS.
C             R1V      SEEKER RANGE TO TARGET IN METERS.
C
C  CN2
C           IR2      STARTING INDEX VALUE OF CN2(I)
```

```
C           (CN2(I), I=IR2,IR2+5)
C             CN2(I)  VALUES OF REFRACTIVE INDEX STRUCTURE
C                     CONSTANT FOR EACH SEGMENT OF RANGE FROM TARGET TO
C                     SEEKER (METERS**(-2/3))
C
C V2
C           IR2     STARTING INDEX VALUE OF V2(I)
C
C             V2(I)   VALUES OF CROSSWIND VELOCITY FOR EACH SEGMENT
C                     OF RANGE FROM TARGET TO SEEKER (M/SEC).
C
C NPPS      NONE      PRINTS TABULAR VALUES OF POWER SPECTRUM VS FREQUENCY
C NPPJ      NONE      PRINTS TABULAR VALUES OF POINTING JITTER VS TIME
C NPAL      NONE      PRINTS TABULAR VALUES OF BOTH POWER SPECTRUM (PS)
C                     AND POINTING JITTER (PJ).
C   **IF THIS CARD IS MISSING NO TABLES WILL BE PRINTED (DEFAULT VALUE).
C
C THE FOLLOWING IDENT RECORD IS ALWAYS REQUIRED.
C
C       GO              SIGNIFIES TO BEGIN EXECUTION FOR THIS DATA SET.
C                       AFTER EXECUTION, ANOTHER SET OF INPUTS MAY BE
C                       READ-IN FOLLOWED BY ANOTHER 'GO' CARD.
C                       ANY VALUES ESTABLISHED FROM PREVIOUS INPUT SETS
C                       TO THE ROUTINE ARE STILL IN EFFECT.  THUS DATA
C                       SUCH AS FROM CARD PARM NEED NOT BE READ AGAIN IF
C                       THERE ARE TO BE NO CHANGES IN THE DATA ASSOCIATED
C                       WITH THAT IDENTIFIER.
C
C*********************************************************************
C     ++ CALLED PROGRAMS ++
C     DESUB
C     FALPH
C     FFT4
C     GAUSS
C     MEANVR
C     RAND
C     SPECT
C     SPREAD
C     THETO
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      COMPLEX RAN
      REAL LAMB
      REAL INR(7),IR1,LABEL(11)
      LOGICAL SETUP
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C     THIS IS A COMPLEX NUMBER  EACH NUMBER TAKES TWO WORDS
      COMMON /M05/RAN(2048)
      COMMON /M01/FR(1025),CN1(20),V1(20),FO(20),RO(20)
      COMMON /LOWEX/PS(1025),V2(20),RR(10)
      DIMENSION CN2(20),PJCHAR(4),PSCHAR(4)
      COMMON /GEOMET/PTS(15),IGEOSW
      EXTERNAL DESUB,FALPH
      DATA SETUP/.TRUE./
      DATA PJCHAR,PSCHAR /4HPJ (,4HRAD*,4H*2/S,4HEC) ,4HPS (,4HRAD*,
     1                   4H*2/H,4HZ)  /
      DATA LABEL/4HGO  ,4H    ,4HPARM,4HCN1 ,4HV1  ,4HDVRV,4HCN2 ,
     1 4HV2  ,4HNPPS,4HNPPJ,4HNPAL/
C
      NPRINT=0
      LAMB=WAVE
C  SET THE SEED FOR THE RANDOM NUMBER GENERATOR
C***  NOTE, THIS SEED IS APPROPRIATE FOR THE RANDOM NUMBER GENERATOR
C***  USED AT THE ATMOSPHERIC SCIENCES LAB.  USERS AT OTHER
C***  INSTALLATIONS WILL NEED TO SUPPLY THEIR OWN RANDOM NUMBER
C***  GENERATOR.
      IF (.NOT.SETUP) GO TO 100
      VARX=735.34829
      VARX=RAND(VARX)
      SETUP=.FALSE.
```

```
      IOPT=1
  100 READ(IOIN,3000) IR1,(INR(I), I=1,7)
      IR2=IFIX(INR(1))
      IF(IR1.EQ.LABEL(1)) GO TO 180
      IF(IR1.EQ.LABEL(3)) GO TO 110
      IF(IR1.EQ.LABEL(4)) GO TO 120
      IF(IR1.EQ.LABEL(5)) GO TO 130
      IF(IR1.EQ.LABEL(6)) GO TO 140
      IF(IR1.EQ.LABEL(7)) GO TO 150
      IF(IR1.EQ.LABEL(8)) GO TO 160
C     PRINTING OPTIONS
      IF(IR1.EQ.LABEL(9)) NPRINT=1
      IF(IR1.EQ.LABEL(10)) NPRINT=2
      IF(IR1.EQ.LABEL(11)) NPRINT=3
      IF(NPRINT.EQ.1.OR.NPRINT.EQ.2.OR.NPRINT.EQ.3) GO TO 100
      WRITE(IOOUT,3001) IR1,(INR(I), I=1,7)
      GOTO 100
C
  110 DIAM = INR(1)
      THET = INR(2)
      TDOT = INR(3)
      RANG = INR(4)
      TIME = INR(5)
      M = IFIX(INR(6))
      IF(M .EQ. 0) M = 512
      GOTO 100
C
  120 DO 125 I=2,7
      IF(INR(I).NE.0.0) GOTO 121
      N1=IR2-1
      GOTO 100
  121 CN1(IR2)=INR(I)
      IR2=IR2+1
      IF(IR2 .GT. 20) GOTO 126
  125 CONTINUE
  126 IR1=IR2-1
      N1=MAX0(N1,IFIX(IR1))
      GOTO 100
C
  130 DO 131 I=2,7
      V1(IR2)=INR(I)
      IR2=IR2+1
      IF(IR2 .GT. 20) GOTO 100
  131 CONTINUE
      GOTO 100
C
  140 IOPT = 2
      D1V = INR(1)
      R1V = INR(2)
      GOTO 100
C
  150 IOPT = 2
      DO 155 I=2,7
      IF(INR(I).NE.0.0) GOTO 151
      N2=IR2-1
      GOTO 100
  151 CN2(IR2) = INR(I)
      IR2=IR2+1
      IF(IR2 .GT. 20) GOTO 156
  155 CONTINUE
  156 IR1=IR2-1
      N2=MAX0(N2,IFIX(IR1))
      GOTO 100
C
  160 IOPT = 2
      DO 161 I=2,7
      V2(IR2) = INR(I)
      IR2=IR2+1
      IF(IR2 .GT. 20) GOTO 100
  161 CONTINUE
```

```
      GOTO 100
C
  180 IF(IGEOSW.NE.1)GO TO 190
      RANG=SQRT((PTS(7)-PTS(1))**2+(PTS(8)-PTS(2))**2+
     +(PTS(9)-PTS(3))**2)
      DISKTM=1000.0
      RANG=RANG*DISKTM
      R1V=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+(PTS(6)-PTS(3))**2)
      R1V=R1V*DISKTM
  190 CONTINUE
      IF (LAMB.LE.14.) GO TO 200
      WRITE (IOOUT,2500) LAMB
      IERR=1
      RETURN
  200 WRITE (IOOUT,3100)
      IF (IOPT.EQ.2) WRITE (IOOUT,3200)
      WRITE (IOOUT,3300) LAMB,DIAM,THET
C  CHANGE WAVELENGTH TO METERS
      LAMB=LAMB/1.0E+6
      IF (IOPT.EQ.2) WRITE (IOOUT,3400) D1V,R1V
      WRITE (IOOUT,3500) TDOT,RANG
      WRITE (IOOUT,3600) TIME
      WRITE (IOOUT,3700) N1
      IF (IOPT.EQ.2) WRITE (IOOUT,3800) N2
      WRITE (IOOUT,3900) M
      WRITE (IOOUT,4000)
      WRITE (IOOUT,4200)
      DO 400 I=1,N1
      WRITE (IOOUT,4300) I,CN1(I),V1(I)
  400 CONTINUE
      IF (IOPT.EQ.1) GO TO 600
      WRITE (IOOUT,4100)
      WRITE (IOOUT,4200)
      DO 500 I=1,N2
      WRITE (IOOUT,4300) I,CN2(I),V2(I)
  500 CONTINUE
C  COMPUTATION OF TIME, FREQUENCY AND SPATIAL INCREMENTS
  600 DELT=TIME/M
      DELF=1./TIME
      DELZ=RANG/FLOAT(N1)
      IF (N2.NE.0) DEL1V=R1V/FLOAT(N2)
      MM=M+M
      M1=M+1
      MM1=MM+1
      MM2=MM+2
      MSQ=SQRT(FLOAT(MM))
      DO 700 I=2,M1
      FR(I)=(I-1)*DELF
  700 CONTINUE
      R2=DIAM/THET
      R=RANG+R2
      DT=THET*RANG
      D2=DIAM+DT
C  COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND
C  NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM LASER
C  DESIGNATOR TO TARGET
      ZI=0.0
      Z1=DELZ/2.
      ROT=0.0
      DEL=0.0
      DO 800 I=1,N1
      ZI=ZI+Z1
      Z1=DELZ
      VEI=V1(I)+TDOT*(ZI-R2)
      RO(I)=16.71*DELZ*CN1(I)*(1.-ZI/RANG)**1.66667/(LAMB*LAMB)
      FO(I)=VEI/(PI*D2*ZI/R)
      ROT=ROT+RO(I)
      DEL=DEL+DELZ*CN1(I)*(RANG-ZI)/RANG
      RO(I)=RO(I)**(-.6)
  800 CONTINUE
```

```
      ROT=ROT**(-.6)
      CALL SPREAD(DIAM,ROT,LAMB,THET,RANG,DRO,THETDL,DDL,THET12,D12,
     1            DTHET,D22)
      WRITE (IOOUT,4500) R2,R
      WRITE (IOOUT,4600) ROT,DRO,THET,DT,THETDL,DDL,THET12,D12,DTHET,D22
      WRITE (IOOUT,4700)
      DO 1000 I=1,N1
 1000 WRITE (IOOUT,4800) I,RO(I),FO(I)
C  COMPUTATION OF ANGLE OF ARRIVAL POWER SPECTRUM OF LASER  DESIGNATOR
      F2=0.
      PS(1)=0.
      DO 1100 J=2,M1
      F=FR(J)
      F1=0.
      CALL SPECT(F,D2,N1,F1,LAMB)
      PS(J)=F1
      IF (IOPT.EQ.1) PS(J)=F1*(D2/DIAM)**2
      F2=F2+PS(J)*DELF
 1100 CONTINUE
      IF (IOPT.EQ.1) WRITE (IOOUT,4400) F2
      F2SQRT=SQRT(F2)
      AJITT=F2SQRT*RANG
      IF (IOPT.EQ.1) WRITE (IOOUT,4900) F2SQRT,AJITT
      IF (IOPT.EQ.1) GO TO 1500
C
      DTV=THET*R1V
C  COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND
C  NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM
C  TARGET TO SEEKER
      ZI=0.
      Z1=DEL1V/2.
      ROT=0.0
      DO 1200 I=1,N2
      ZI=ZI+Z1
      Z1=DEL1V
      VEI=V2(I)+TDOT*(R1V-ZI)
      RO(I)=16.71*DEL1V*CN2(I)*(ZI/R1V)**1.66667/(LAMB*LAMB)
      ROT=ROT+RO(I)
      RO(I)=RO(I)**(-.6)
      FO(I)=VEI/(PI*D1V*ZI/R1V)
 1200 CONTINUE
      ROT=ROT**(-.6)
      CALL SPREAD(D22,ROT,LAMB,THET,R1V,DRO,THETDL,DDL,THET12,D12,DTHET,
     1            D22V)
      WRITE (IOOUT,4550)
      WRITE (IOOUT,4600) ROT,DRO,THET,DTV,THETDL,DDL,THET12,D12,DTHET,
     1                   D22V
      WRITE (IOOUT,4700)
      DO 1300 I=1,N2
 1300 WRITE (IOOUT,4800) I,RO(I),FO(I)
C  COMPUTATION OF TURBULENCE INDUCED POINTING JITTER POWER SPECTRUM
C  FROM TARGET SPOT TO LASER SEEKER,  COMPUTATION OF TOTAL POWER
C  SPECTRUM FROM LASER DESIGNATOR TO SEEKER AND POWER SPECTRUM VARIANCE
      CALL THETO(THETAO,FALPH,CN2,D1V,R1V,N2)
      F2=0.
      DO 1400 J=2,M1
      F=FR(J)
      F1=0.
      CALL SPECT(F,D1V,N2,F1,LAMB)
      PS(J)=PS(J)+F1/(1.+(D2/(R1V*THETAO))**2)
      PS(J)=PS(J)*(D2/D1V)**2
      F2=F2+PS(J)*DELF
 1400 CONTINUE
      WRITE (IOOUT,4400) F2
 1500 IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5000) DELF
      FREQ=0.0
      FINC=DELF*10.0
      IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5600)
      IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PSCHAR
      K=1
```

```
      L=10
 1600 IF (L.GT.M1) L=M1
      IF(NPRINT.EQ.1.OR.NPRINT.EQ.3)WRITE (IOOUT,2800)FREQ,(PS(J),J=K,L)
      FREQ=FREQ+FINC
      K=L+1
      L=K+9
      IF (K.LE.M1) GO TO 1600
      DO 2400 L=1,IOPT
      IF (L.EQ.1) WRITE (IOOUT,5100)
      IF (L.EQ.2) WRITE (IOOUT,5200)
C   GENERATION OF RANDOM SEQUENCE HAVING SAME POWER SPECTRUM VARIANCE
C   AS INDUCED BY TURBULANCE.  ADD SYMMETRIC TERMS FOR NEGATIVE
C   FREQUENCIES.  COMPUTE MEAN AND VARIANCE OF RANDOM ARRAY
      RAN(1)=(0.,0.)
      DO 1700 I=2,M1
      MMM=MM2-I
C   GENERATE RANDOM NUMBER WITH ROUTINE GAUSS
C       NORMAL DISTRIBUTION
C         MEAN = 0.0
C         STANDARD DEVIATION = 1.0
C       REAL PART = RANDOM NUMBER
C       IMAG PART = 0
      RAN(I)=CMPLX(GAUSS(12,0.0,1.0),0.0)*SQRT(PS(I)/DELT)
      RAN(MMM)=RAN(I)
 1700 CONTINUE
C   CALCULATE POWER OF 2 ,NPOW, FOR FFT4 SINCE ARRAY PASSED TO
C   FFT4 MUST HAVE SIZE THAT IS A POWER OF 2.  (NOTE LN(2)=0.693147.)
      NPOW=IFIX(ALOG(FLOAT(MM))/0.693147)
      NMAX=2**NPOW
      IF (MM.EQ.NMAX) GO TO 1900
C   IF MM IS NOT A POWER OF 2 THEN RESET THE REST OF ARRAY RAN
      NPOW=NPOW+1
      ISTART=MM+1
      NMAX=2**NPOW
      DO 1800 I=ISTART,NMAX
      RAN(I)=(0.0,0.0)
 1800 CONTINUE
C   COMPUTE AND WRITE MEAN AND VARIANCE OF RANDOM ARRAY
 1900 WRITE (IOOUT,5300)
      CALL MEANVR(1,M)
C   FAST FOURIER TRANSFORM RANDOM ARRAY
C   CDC ROUTINE CALL
C             CALL FFT(RAN,MM,+1)
      CALL FFT4(1.0,RAN,NPOW,NMAX)
      DO 2100 I=1,MM
      RAN(I)=RAN(I)/MSQ
 2100 CONTINUE
C   COMPUTE AND WRITE MEAN AND VARIANCE OF TIME SEQUENCE.
      WRITE (IOOUT,5500)
      CALL MEANVR(M1,MM)
C   WRITE TRANSFORMED ARRAY VALUES CORRESPONDING TO TIME VALUES
C   OF POINTING JITTER FOR ONE DIRECTION.
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5400) DELT
      DTIME=DELT
      TINC=DELT*10.0
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5700)
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PJCHAR
      DO 2300 I1=M1,MM,10
      DO 2200 I2=1,10
      I3=I1+I2-1
      RR(I2)=REAL(RAN(I3))
 2200 CONTINUE
      IF(NPRINT.EQ.2.OR.NPRINT.EQ.3)WRITE(IOOUT,2800)DTIME,
     1     (RR(I),I=1,10)
      DTIME=DTIME+TINC
 2300 CONTINUE
 2400 CONTINUE
C
C
 2500 FORMAT (1X,100(1H*),/,13H WAVELENGTH (,F10.3,7H) GREAT,
```

```
     1          40HER THAN 14 MICRONS: CONTROL RETURNED TO ,
     2          21HMAIN FROM TURBULENCE.,/,1X,100(1H*))
2700 FORMAT (5F16.6)
2800 FORMAT (E10.4,10E12.4)
3000 FORMAT (A4,6X,7E10.4)
3001 FORMAT(1H0,20X,45HTHE FOLLOWING ID-FIELD NOT RECOGNIZED BY TURB,/,
     1 1X,A4,6X,7E10.4)
3100 FORMAT (1H1,36H  CALCULATION OF POWER SPECTRUM AND
     1          11HTURBULENCE 29HINDUCED POINTING JITTER OF A
     2          23HLASER TARGET DESIGNATOR)
3200 FORMAT (1H+,101X,10HAND SEEKER)
3300 FORMAT (1H0,41X,30HLASER WAVELENGTH (MICROMETERS),8X,
     1          F10.4,//,42X,26HDESIG. APERTURE DIAMETER (
     2          7HMETERS),5X,F10.6,//,42X,18HBEAMSPREAD ANGLE (
     3          8HRADIANS),12X,F10.6)
3400 FORMAT (1H0,41X,33HSEEKER APERTURE DIAMETER (METERS),5X,
     1          F10.6,//,42X,36HRANGE FROM TARGET TO SEEKER (METERS),
     2          2X,F10.2)
3500 FORMAT (1H0,41X,24HBEAM SLUE RATE (RAD/SEC),14X,F10.6,
     1          //,42X,26HDESIGNATION RANGE (METERS),12X,
     2          F10.2)
3600 FORMAT (1H0,41X,26HDURATION OF TEST (SECONDS),12X,
     1          F10.4)
3700 FORMAT (1H0,41X,30HTOTAL DESIGNATOR PATH SEGMENTS,15X,
     1          I3)
3800 FORMAT (1H0,41X,26HTOTAL SEEKER PATH SEGMENTS,19X,I3)
3900 FORMAT (1H0,41X,27HTOTAL FREQUENCIES FOR WHICH,/,42X,
     1          22H  POWER SPECTRUM IS TO,/,42X,
     2          15H  BE CALCULATED,27X,I4)
4000 FORMAT (1H0,///,26X,29H  VALUES OF REFRACTIVE INDEX
     1          37HSTRUCTURE CONSTANT AND WIND SPEED IN
     2          15HDESIGNATOR PATH)
4100 FORMAT (1H0,///,26X,29H  VALUES OF REFRACTIVE INDEX
     1          37HSTRUCTURE CONSTANT AND WIND SPEED IN
     2          11HSEEKER PATH)
4200 FORMAT (1H0,62X,5HCN**2,9X,9HWINDSPEED,16X,/,42X,
     1          11HSEGMENT NO.,5X,15H(METER**(-2/3)),3X,
     2          11H(METER/SEC),//)
4300 FORMAT (1H ,46X,I2,10X,E12.6,5X,F10.2)
4400 FORMAT (1H0,//,36H THE VARIANCE OF THE POWER SPECTRUM
     1          3HIS ,E12.4)
4500 FORMAT (1H1,//,56X,20HDESIGNATOR TO TARGET,//,38X,
     1          //,24X,41HVIRTUAL POINT SOURCE TO APERTURE DISTANCE,
     2          28X,F10.5,9H (METERS),//,24X,22HDISTANCE FROM VIRTUAL
     3          22HPOINT SOURCE TO TARGET,25X,F10.5,9H (METERS))
4550 FORMAT (1H1,//,58X,16HTARGET TO SEEKER,//)
4600 FORMAT (1H0,23X,27HINTEGRATED COHERENCE LENGTH,42X,F10.6,
     1          9H (METERS),//,24X,36HDIAMETER/INTEGRATED COHERENCE LENGTH
     2          ,33X,F10.6,//,24X,31HTRANSMITTER-INDUCED BEAM SPREAD
     3          13X,E12.5,10H (RADIANS),3X,F10.6,9H (METERS),
     4          //,24X,31HDIFFRACTION-LIMITED BEAM SPREAD,13X,E12.5,
     5          10H (RADIANS),3X,F10.6,9H (METERS),//,24X,
     6          38HDIFFRACTION AND TURBULENCE BEAM SPREAD,6X,E12.5,
     7          10H (RADIANS),3X,F10.6,9H (METERS),///,24X,6HTOTAL
     8          19HEFFECTIVE BEAM SIZE,19X,E12.5,10H (RADIANS),3X,F10.6,
     9          9H (METERS),//)
4700 FORMAT (1H0,34X,11HSEGMENT NO.,9X,16HCOHERENCE LENGTH,
     1          5X,27H REFERENCE FREQUENCY(HERTZ),//)
4800 FORMAT (1H ,38X,I2,10X,F16.6,10X,F16.6)
4900 FORMAT (1H0,19H RMS SPOT JITTER = ,E10.4,10H RAD, OR =
     1          ,E10.4,7H METERS,/)
5000 FORMAT (1H1,1X,40X,31H CALCULATED POWER SPECTRUM VS.
     1          9HFREQUENCY,/,47X,16H AT INTERVALS OF,F6.3,6H HERTZ)
5100 FORMAT (1H0,37H OUTPUT FOR DESIGNATOR TO TARGET PATH)
5200 FORMAT (1H0,33H OUTPUT FOR TARGET TO SEEKER PATH)
5300 FORMAT (1H0,35X,35H  MEAN AND VARIANCE OF RANDOM ARRAY
     1          ,//)
5400 FORMAT (1H1,49X,34HVALUES OF POINTING JITTER VS. TIME /
     1          52X,16H AT INTERVALS OF,F8.4,4H SEC)
5500 FORMAT (1H0,//,35X,28H  MEAN AND VARIANCE OF TIME
```

```
     1        8HSEQUENCE//)
 5600 FORMAT (1H ,10H BEGINNING,/,10H FREQ (HZ))
 5700 FORMAT (1H ,2X,5H TIME,/,2X,6H (SEC))
 5800 FORMAT (1H+,T13,46(1H-),1H ,4A4,55(1H-),/)
      RETURN
      END
```

```
      FUNCTION DESUB(X,DRO)
C   FUNCTION USED FOR INTEGERATION  F(X)
      FACTR=3.44*(DRO*X)**1.666667*(1.-X**0.333333)
      IF(FACTR.LT.160.) GO TO 10
      DESUB=0.
      GO TO 20
   10 ARCCX=ATAN2(SQRT(1.-X**2),X)
      DESUB=X*((ARCCX-X*(1.-X**2)**.5)*EXP(-FACTR))
   20 RETURN
      END
```

```
      FUNCTION FALPH(XI1)
      IF (XI1.GE..5623) GO TO 100
      FALPH=10.66*((XI1)**2)
      GO TO 900
  100 IF (XI1.GE.1.0) GO TO 200
      FALPH=4.025*XI1-.00659
      GO TO 900
  200 IF (XI1.GE.1.778) GO TO 300
      FALPH=1.8547*XI1+2.164
      GO TO 900
  300 IF (XI1.GE.3.162) GO TO 400
      FALPH=.8475*XI1+3.955
      GO TO 900
  400 IF (XI1.GE.5.623) GO TO 500
      FALPH=.391*XI1+5.3977
      GO TO 900
  500 IF (XI1.GE.10.) GO TO 600
      FALPH=.1814*XI1+6.578
      GO TO 900
  600 IF (XI1.GT.31.62) GO TO 700
      FALPH=.0534*XI1+7.95
      GO TO 900
  700 IF (XI1.GT.1000.) GO TO 800
      FALPH=7.8*(XI1**.06)
      GO TO 900
  800 FALPH=11.97
  900 RETURN
      END
```

```
      SUBROUTINE FFT4(SIGN,X,NPOW,NMAX)
C
C     COOLEY-TUKEY METHOD OF FOURIER TRANSFORM
C     INCLUDES SINE COSINE COMPUTATION AND
C     REARRANGING DATA ACCORDING TO REVERSE BIT ADDRESSES
C
C     SIGN = FOURIER DIRECTION TRANSFORM FLAG
C            -1. FOR DIRECT TRANSFORM, TO COEFFICIENTS FROM SERIES
C             1. FOR INVERSE TRANSFORM, TO SERIES FROM COEFFICIENTS
C
C     X      = LOC. OF FOURIER TRANSFORM BLOCK
C
C     NPOW = POWER OF 2 TO OBTAIN NMAX
C
C     NMAX = LENGTH OF BLOCK X
C
      COMPLEX X,CXCS,HOLD,XA
      DIMENSION CS(2),MSK(13)
      DIMENSION X(1)
      EQUIVALENCE (CXCS,CS)
      ZZ=6.283185306*SIGN/FLOAT(NMAX)
      MSK(1)=NMAX/2
      DO 100 I=2,NPOW
      MSK(I)=MSK(I-1)/2
  100 CONTINUE
      NN=NMAX
      MM=2
C     LOOP OVER NPOW LAYERS
      DO 800 LAYER=1,NPOW
      NN=NN/2
      NW=0
      DO 700 I=1,MM,2
      II=NN*I
C           CXCS = CEXP(2*PI*NW*SIGN/NMAX)
      W=FLOAT(NW)*ZZ
      CS(1)=COS(W)
      CS(2)=SIN(W)
C     COMPUTE ELEMENTS FOR BOTH HALFS OF EACH BLOCK
      DO 200 J=1,NN
      II=II+1
      IJ=II-NN
      XA=CXCS*X(II)
      X(II)=X(IJ)-XA
      X(IJ)=X(IJ)+XA
  200 CONTINUE
C     BUMP UP SERIES BY 2
C     COMPUTE REVERSE ADDRESS
      DO 400 LOC=2,NPOW
      LL=NW-MSK(LOC)
      IF (LL) 500,600,300
  300 NW=LL
  400 CONTINUE
  500 NW=MSK(LOC)+NW
      GO TO 700
  600 NW=MSK(LOC+1)
  700 CONTINUE
      MM=MM*2
  800 CONTINUE
C     DO FINAL REARRANGEMENT
      NW=0
      DO 1600 I=1,NMAX
      NW1=NW+1
      HOLD=X(NW1)
      IF (NW1-I) 1100,1000,900
  900 X(NW1)=X(I)
 1000 X(I)=HOLD
C     BUMP UP SERIES BY 1
C     COMPUTE REVERSE ADDRESS
 1100 DO 1300 LOC=1,NPOW
      LL=NW-MSK(LOC)
```

```
      IF (LL) 1400,1500,1200
 1200 NW=LL
 1300 CONTINUE
 1400 NW=MSK(LOC)+NW
      GO TO 1600
 1500 NW=MSK(LOC+1)
 1600 CONTINUE
      IF (SIGN) 1900,1900,1700
 1700 PTS=NMAX
      DO 1800 I=1,NMAX
      X(I)=X(I)/PTS
 1800 CONTINUE
 1900 RETURN
      END
```

```
      FUNCTION GAUSS(N,XBAR,SIGMA)
C  GENERATE RANDOM NUMBERS WITH NORMAL DISTRIBUTION
C    MEAN = XBAR
C    STANDARD DEVIATION = SIGMA
      DATA NN /0/
      IF (NN.GT.0) GO TO 1
      NN=1
      C=0.
1     CONTINUE
      X=0.0
      IF (C.EQ.0.0) C=735.34829
      DO 100 J=1,N
      C=RAND(C)
      X=X+C
  100 CONTINUE
      XN=N
      X=SQRT(12.0/XN)*(X-0.5*XN)
      GAUSS=SIGMA*X+XBAR
      RETURN
      END
```

```
      SUBROUTINE MEANVR(N1,N2)
C
C  COMPUTE AND WRITE MEAN AND VARIANCE OF COMPLEX ARRAY
C  OVER SOME RANGE OF THE ARRAY.
C
C      ++ INPUT ++
C
C  N1 = STARTING INDEX OF RANGE
C
C  N2 = ENDING INDEX OF RANGE
C
C      ++ COMMON ++
C
C  RAN = COMPLEX ARRAY CONTAINING DATA
C
C      ++ OUTPUT ++
C
C  MEAN1 = MEAN OF REAL PART
C
C  MEAN2 = MEAN OF IMAGINARY PART
C
C  VAR1 = VARIANCE OF REAL PART
C
C  VAR2 = VARIANCE OF IMAGINARY PART
C
      COMPLEX RAN
      REAL MEAN1,MEAN2
C         THIS IS A COMPLEX NUMBER  EACH NUMBER TAKES TWO WORDS
      COMMON /M05/RAN(2048)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      MEAN1=0.
      MEAN2=0.
      VAR1=0.
      VAR2=0.
      DO 100 I=N1,N2
      MEAN1=REAL(RAN(I))+MEAN1
      MEAN2=AIMAG(RAN(I))+MEAN2
      VAR1=(REAL(RAN(I)))**2+VAR1
      VAR2=(AIMAG(RAN(I)))**2+VAR2
  100 CONTINUE
      MEAN1=MEAN1/FLOAT(N2-N1)
      MEAN2=MEAN2/FLOAT(N2-N1)
      VAR1=VAR1/FLOAT(N2-N1)
      VAR2=VAR2/FLOAT(N2-N1)
      WRITE (IOOUT,200) MEAN1,MEAN2
      WRITE (IOOUT,300) VAR1,VAR2
      RETURN
C
  200 FORMAT (35X,20H MEAN OF REAL PART =,E12.5,10H, MEAN OF
     1        11HIMAG PART =,E12.5)
  300 FORMAT (35X,20H VAR. OF REAL PART =,E12.5,10H, VAR. OF
     1        11HIMAG PART =,E12.5)
      END
```

```
      SUBROUTINE SPECT(F,D2,N,F1,LAMB)
C  THIS ROUTINE GENERATES THE APPROXIMATE FUNCTION
C     G<ALPHA> (F/F<SUB 0,I>)
C  USED IN THE POWER SPECTRUM OF ANGLE-OF-ARRIVAL EQUATION
      REAL LAMB
      COMMON /M01/FR(1025),CN(20),V1(20),F0(20),R0(20)
      FACT=1.32E-2*(LAMB/D2)**2
      F1=0.
      DO 100 I=1,N
      IF (F.LE..332*F0(I)) G=1.
      IF (F.GT..332*F0(I)) G=1.12-.361*F/F0(I)
      IF (F.GE.3.10*F0(I)) G=0.
      F1=F1+FACT*((D2/R0(I))**5/(F*F*F0(I)))**.33333*G
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SPREAD(DIAM,ROT,WAVE,THET,RANG,DRO,THETDL,DDL,THET12,
     1                  D12,DTHET,DTOT)
C
C   COMPUTATION OF BEAM SPREAD ANGLE DUE TO DIFFRACTION AND DIFFRACTION
C   AND TURBULENCE AND SPOT DIAMETER ON TARGET (OR SEEKER).
C
      DRO=DIAM/ROT
C   DO 1/2 SIMPSON RULE INTEGRATION
      VARX=0.0
      DELTAX=0.01
      RDRO=DESUB(VARX,DRO)/2.0
      DO 100 I=1,100
      VARX=VARX+DELTAX
      RDRO=RDRO+DESUB(VARX,DRO)
  100 CONTINUE
      RDRO=(RDRO-DESUB(VARX,DRO)/2.0)*DELTAX
      RDRO=1.0/(SQRT(5.092958*(DRO)**2*RDRO))
      THETDL=1.128*WAVE/DIAM
      DDL=THETDL*RANG
      THET12=THETDL*DRO*RDRO
      D12=THET12*RANG
      DTHET=SQRT(THET12**2+THET**2)
      DTOT=DIAM+DTHET*RANG
      RETURN
      END
```

```
      SUBROUTINE THETO(THETAO,FALPH,CN2,D1V,R1V,N2)
C   ++ CALLED FUNCTIONS ++
C     FALPH
      DIMENSION CN2(20)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DEL1V=R1V/FLOAT(N2)
C     CALCULATE D1INF
      D1INF=0.
      S1=DEL1V/2.
      S=0.
      D1V3=(D1V)*(-.3333)
      DO 100 I=1,N2
      S=S+S1
      D1INF=DEL1V*CN2(I)*((S/R1V)**1.6667)+D1INF
      S1=DEL1V
  100 CONTINUE
      D1INF=0.5*11.97*D1V3*D1INF
C   INITIAL ESTIMATE FOR THETAO
C     WRITE (IOOUT,500) D1INF
      THETAO=1.E-4
  200 XI1=0.
      DITHE=0.
      S=0.
      S1=DEL1V/2.
      DO 300 I=1,N2
      S=S+S1
      XI1=THETAO*(R1V-S)/D1V
      D1THE=DEL1V*CN2(I)*((S/R1V)**1.6667)*FALPH(XI1)+D1THE
      S1=DEL1V
  300 CONTINUE
      D1THE=D1THE*D1V3
      IF (ABS((D1INF-D1THE)/D1INF).LT..001) GO TO 400
      THETAO=THETAO*(1.+.5*(D1INF-D1THE)/D1INF)
C     WRITE (IOOUT,500) THETAO,D1THE
      GO TO 200
  400 CONTINUE
C     WRITE (IOOUT,500) D1THE
      RETURN
C
  500 FORMAT (2E16.8)
      END
```

```
      SUBROUTINE BASCAT(WAVE,EXCO,IERR)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C     THIS VERSION OF BASCAT (20 SEP 81) DIFFERS FROM THE EOSAEL 80
C     VERSION IN ITS INTERNAL PROGRAM STRUCTURE AND OUTPUT CAPABILITY.
C     INPUT FORMATTING HAS NOT BEEN CHANGED. BRIEFLY, THE INTERNAL
C     STRUCTURAL CHANGES CONSIST OF THE FOLLOWING :
C
C       (A) SUBROUTINE THIT HAS BEEN ELIMINATED. THE FUNCTIONS WHICH
C           IT ONCE PERFORMED HAVE BEEN CONSOLIDATED INTO SUBROUTINE
C           START.
C       (B) A NEW LIDAR BIASING ALGORITHM HAS BEEN INSERTED.
C       (C) DIRECT BEAM (I.E., UNSCATTERED) COMPUTATIONS IN
C           SUBROUTINE START HAVE BEEN REVISED.
C       (D) SUBROUTINE CONV HAS BEEN MODIFIED SO THAT THE DIFFER-
C           ENCE OF TWO NUMBERS RETAINS MORE SIGNIFICANT DIGITS.
C       (E) NUMERICAL CHECKS FOR IMPROPER ARGUMENTS OF FUNCTIONS
C           (DIVISIONS, SQUARE ROOTS, LOGARITHMS, ETC.) HAVE BEEN
C           REVISED.
C       (F) ARGUMENT LISTS OF A FEW COMMON BLOCKS HAVE BEEN CHANGED.
C       (G) WRITE STATEMENTS FOR OUTPUTTING BASCAT RESULTS TO A USER-
C           DEFINED PLOT FILE (NPLOTU) HAVE BEEN INCLUDED. THESE STATE-
C           MENTS MUST BE UNCOMMENTED IN ORDER TO ACTIVATE THEM.
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C     THE BASCAT MODULE ALONE USES THE FOLLOWING SUBROUTINES:
C       BKWD - CONTAINS BACKWARD SCATTERING ALGORITHM
C       CONV - CONVOLVES IMPULSE RESPONSE WITH SQUARE PULSE
C       ELM - DETERMINES BIASING DISTANCES
C       FIND - DETERMINES INTERPOLATED PHASE FUNCTION VALUE
C       FWRD - FIRST ORDER SCATTERING ALGORITHM
C       GAS - DETERMINES MONTE CARLO SCATTERING ANGLES FOR TRAVERSES
C       GMAX - DETERMINES MAXIMUM OF AN INPUT ARRAY
C       MATRX - GENERATES ROTATION MATRICES
C       ROTAT - ROTATES VECTORS FROM ONE COORDINATE SYSTEM TO ANOTHER
C       SMOOZ - DETERMINES START OF TRAILING ZEROS IN INPUT ARRAY
C       START - INITIATES PHOTON TRAJECTORIES
C       TRAVRS - MOVES PHOTONS BETWEEN SCATTERING POINTS AND FINDS OB-
C                SERVED POWER CONTRIBUTIONS AT THOSE POINTS
C       USCA - SELECTS RANDOM ANGLES WEIGHTED BY PHASE FUNCTION
C
C     TWO SUBROUTINES SHARED BY BASCAT WITH OTHER EOSAEL 80 MODULES
C     ARE THE FOLLOWING :
C      PFUNC - SELECTS AND RENORMALIZES PHASE FUNCTION DATA FROM
C              EOSAEL 80 DATA BASE
C       RAND - RANDOM NUMBER GENERATOR (GENERATES UNIFORM DISTRIBUTION
C              OF RANDOM NUMBERS BETWEEN 0 AND 1)
C
C** NOTE** THE FOLLOWING ROUTINES UTILIZE THE RANDOM NUMBER GENERATOR
C          WHICH IS INVOKED AS FUNCTION 'RAND(SEED)' :
C
C            (A) BKWD - 3 OCCURRENCES OF FUNCTION RAND
C            (B) FWRD - 2 OCCURRENCES
C            (C) GAS - 3 OCCURRENCES
C            (D) START - 2 OCCURRENCES
C            (E) TRAVRS - 2 OCCURRENCES
C
C          USERS OF OTHER (NON-HP) COMPUTER SYSTEMS MUST REPLACE
C          FUNCTION RAND(SEED) WITH A UNIFORM RANDOM NUMBER GENERATOR
C          WHICH WORKS FOR THEIR SYSTEMS. THE RANDOM NUMBER SEED IS
C          INITIALIZED IN SUBROUTINE BASCAT. THIS AND ALL SUBSEQUENT
C          VALUES OF THE RANDOM NUMBER SEED ARE PASSED VIA COMMON
C          BLOCK 'RNDM'. SHOULD THE RANDOM NUMBER SEED USED HERE
C          (735.34829) BE INAPPROPRIATE FOR THE USER'S SYSTEM, IT IS
C          SUGGESTED THAT THE INITIALIZATION USED BELOW (SEED0=
C          735.34829) BE CHANGED TO AN APPROPRIATE VALUE.
C
C     SUBSTANTIAL MODIFICATIONS HAVE BEEN MADE TO THE FOLLOWING
C     SUBROUTINES PRESENT IN EOSAEL 80 :
C       CONV
```

```
C       TRAVRS
C       START
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      COMMON/RNDM/SEED
      COMMON /GEOMET/PTS(15),IGEOSW
      COMMON/ALL/AT,BT,CT,BLIM
      COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
     +NTMAX,NSCAM,KMAX,LMAX,LMM1
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
     +BE(16),SINGWV,PF(65),LLMAX
      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
     *ALB(2),ZG,DMAX
      COMMON/FGEL/XA(3),D,NSCA
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
      COMMON/CONB/X(100),Y(100)
      COMMON/HIT/UDS,THSP,RS(3,3),XS(3),DSA,XV(3)
      DIMENSION NM(10),TPU(7),SS(3)
      DIMENSION IAL(7),DAT(7),IOR(10)
      DATA IAL/2HPA  ,2HSO  ,2HDE  ,2HCL  ,2HGR  ,2HPU  ,2HGO  /
      DATA IZERO/0/
C*************************************************************
C     THIS SUBROUTINE CALCULATES STEADY STATE AND TIME-DEPENDENT DIRECT
C     AND MULTIPLY SCATTERED POWER INTO A DETECTOR BY AN ELLIPSOIDAL
C     AEROSOL CLOUD WITH GROUND PLANE, FOR A LASER SOURCE.  THE DETECTOR
C     AND SOURCE MAY HAVE ANY LOCATIONS, LOOK ANGLES, AND CONE OF VIEW/
C     BEAM SPREAD/WAVELENGTH.  THE AEROSOL CLOUD MAY HAVE ANY ORIENTA-
C     TION, SIZE, AND SCATTERING PHASE FUNCTION (ARBITRARY NORMALIZA-
C     TION), IN A COORDINATE SYSTEM WITH ORIGIN AT THE CLOUD CENTER,
C     WITH Z-AXIS VERTICAL, X-AXIS EAST, AND Y-AXIS NORTH.  THE GROUND
C     PLANE, ASSUMED AN ISOTROPIC REFLECTOR, MAY HAVE ANY ALBEDO, AND
C     MAY OR MAY NOT INTERSECT THE AEROSOL CLOUD.
C*****************************************************************
C
C** INPUT DATA CARDS ARE READ IN AN ORDER-INDEPENDENT MANNER, WITH
C** A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH RECORD. DATA
C** ON EACH CARD IS READ IN UNDER THE FOLLOWING FORMAT :
C** (A4,1X,7(E9.4,1X)). NOTE THAT INTEGER VARIABLES IN THE PROGRAM
C** MUST BE INPUT AS REAL NUMBERS IN THIS INPUT SCHEME ... THEY ARE
C** LATER FIXED TO THE INTEGER TYPE.
C
C------------------------------------------------------------
C     CARD IDENTIFIER : PART
C     VARIABLES READ : N1,N2,ITIME
C       N1=NUMBER OF PARTIAL OUTPUTS DESIRED, FOR A GIVEN RUN
C       N2=NUMBER OF PHOTONS TO BE USED FOR EACH PARTIAL CALCULATION
C
C** NOTE ** FOR CERTAIN DETECTOR CONDITIONS, AND WITH NORMAL BIASING,
C           AS MANY AS 100,000 PHOTONS MAY BE NEEDED TO OVERCOME LARGE
C           STATISTICAL FLUCTUATIONS IN FIRST ORDER SCATTERING RETURNS.
C           SUCH CONDITIONS ARE DEFINED BY THE FOLLOWING CHARACTERISTICS
C       (A) THE DETECTOR IS IN A MONOSTATIC LIDAR CONFIGURATION.
C       (B) THE DETECTOR IS WITHIN 10 METERS OF THE CLOUD (OR IS INSIDE
C           OF IT).
C       (C) THE CLOUD IS NOT OPTICALLY THICK ALONG THE LOOK DIRECTION
C           (OPTICAL DEPTHS LESS THAN 10).
C*** IN ORDER TO ATTAIN MORE RAPID CONVERGENCE OF FIRST ORDER RETURN
C    POWER UNDER THE ABOVE CONDITIONS, A DIFFERENT BIASING SCHEME IS
C    USED. THE SPECIFIC CONDITIONS WHICH TRIGGER THIS ALTERNATE MODE
C    ARE THE FOLLOWING :
C       (A) THE DOT PRODUCT OF THE SOURCE APERTURE SURFACE NORMAL AND
C           THE DETECTOR APERTURE SURFACE NORMAL IS GREATER THAN 0.99.
C       (B) THE DISTANCE OF THE DETECTOR APERTURE FROM THE NEAREST
C           CLOUD SURFACE (AS SEEN ALONG THE DETECTOR NORMAL) IS LESS
C           THAN OR EQUAL TO 10 METERS (0.01 KM).
C       (C) THE SEPARATION OF SOURCE AND DETECTOR APERTURE CENTERS IS
C           LESS THAN OR EQUAL TO 10 TIMES THE DETECTOR APERTURE
C           RADIUS.
```

```
C*** THE MAIN DIFFERNCES WHICH WILL BE OBSERVED BETWEEN THE NORMAL AND
C    ALTERNATE BIASING MODES ARE THE FOLLOWING :
C       (A) THE FIRST ORDER RETURN POWER IN THE EARLIEST TIME BOX WILL
C           BE EXTREMELY STABLE IN THE ALTERNATE MODE.
C       (B) ONLY THE EARLIEST FIRST ORDER SCATTERING RESULTS WILL BE
C           STRONGLY AFFECTED.
C       (C) LATER FIRST ORDER RETURNS AND ALL HIGHER ORDER RETURNS WILL
C           HAVE NEGLIGIBLY POORER CONVERGENCE.
C*** IN ORDER TO OBSERVE CONVERGENCE OF THE FIRST ORDER STEADY-STATE
C    RETURN POWER TOWARD A STABLE VALUE, IT IS SUGGESTED THAT THE USER
C    UTILIZE THE PARTIAL OUTPUT OPTION. AS AN EXAMPLE, IF 10,000
C    PHOTONS ARE REQUIRED, SET N1=10 AND N2=1,000. THIS SELECTION WILL
C    RUN 10 X 1,000 = 10,000 PHOTONS AND WILL OUTPUT RETURN POWER
C    RESULTS AFTER EACH BATCH OF 1,000 PHOTONS.
C
C     ITIME=OVERALL RUN NUMBER FOR THIS SET OF PARAMETERS. FOR EXAMPLE,
C           ITIME=1 MEANS THE FIRST RUN, ITIME=3 MEANS THAT THE RESULTS
C           OF THE TWO PREVIOUS RUNS WILL BE COMBINED WITH THIS 3RD RUN
C----------------------------------------------------------------
C     CARD IDENTIFIER : SORC
C     VARIABLES READ : XS(1),XS(2),XS(3),THES,PHIS,ASMM
C      (XS(K),K=1,3)=SOURCE XYZ COORDINATES(KM)
C      THES,PHIS=(POLAR,AZIMUTHAL)ANGLES(DEG) OF SOURCE BEAM AXIS
C      ASMM=RADIUS OF SOURCE APERTURE(MM).  THE SOURCE BEAM SPREAD ANGLE
C           'THSP' IS TAKEN BY THE PROGRAM AS THE DIFFRACTION LIMIT FOR
C           THIS APERTURE.  IF YOU SET ASMM=0., THE PROGRAM PUTS THSP=0.
C      IF YOUR SOURCE PHOTONS WOULD NOT INTERSECT THE CLOUD OR THE
C      GROUND, THE SUBROUTINE NOTIFIES YOU, AND RETURNS.
C----------------------------------------------------------------
C     CARD IDENTIFIER : DETR
C     VARIABLES READ : XD(1),XD(2),XD(3),THED,PHID,THEV,ACM
C      (XD(K),K=1,3)=DETECTOR XYZ COORDINATES(KM)
C      THED,PHID=(POLAR,AZIMUTHAL)ANGLES(DEG) OF DETECTOR LOOK AXIS
C      THEV=DETECTOR CONE OF VIEW HALF-ANGLE(DEG)
C      ACM=RADIUS OF DETECTOR DISK(CM)
C     IF YOUR DETECTOR POINTS SKYWARD, OR IF NEITHER YOUR SOURCE NOR
C     YOUR DETECTOR LOOK INTO THE CLOUD, THE SUBROUTINE NOTIFIES YOU,
C     AND RETURNS.
C----------------------------------------------------------------
C     CARD IDENTIFIER : CLDS
C     VARIABLES READ : A(1),A(2),A(3),THE,PHE,PSE,ISO
C      (A(K),K=1,3)=ELLIPSOIDAL CLOUD PRINCIPAL HALF-AXES(KM)
C      THE,PHE,PSE=ELLIPSOID EULER ANGLES(DEG), WHERE 'PHE' IS THE FIRST
C        ROTATION, ABOUT THE Z-AXIS, 'THE' IS THE NEXT, ABOUT THE NEW
C        Y-AXIS, 'PSE' IS THE LAST, ABOUT THE NEW Z-AXIS.
C      ISO=AEROSOL IDENTIFICATION NUMBER, TO COMPARE WITH THE 'ID'
C         PARAMETER.
C----------------------------------------------------------------
C     CARD IDENTIFIER : GRND
C     VARIABLES READ : ZG,ALBG
C      ZG=Z-COORDINATE OF GROUND PLANE(KM)
C      ALBG=GROUND PLANE REFLECTIVITY 0.<ALBG<1.
C     IF YOUR ZG IS SUCH THAT THE GROUND PLANE IS ENTIRELY ABOVE THE
C     CLOUD, THE SUBROUTINE INFORMS YOU, AND RETURNS.  IF YOUR ZG IS
C     NEGATIVE, AND SO LARGE THAT NO GROUND REFLECTIONS WILL RETURN TO
C     THE DETECTOR WITHIN THE TIME LIMIT SET BY THE SUBROUTINE, THE SUB-
C     ROUTINE SETS ALBG=0.  IF YOUR SOURCE IS BELOW THE GROUND PLANE,
C     THE SUBROUTINE PUTS THE SOURCE ON THE GROUND PLANE, AT YOUR XY
C     COORDINATES, AND NOTIFIES YOU.
C----------------------------------------------------------------
C     CARD IDENTIFIER : PULS
C     VARIABLES READ : TPU(1),TPU(2),...,TPU(7)
C      (TPU(J),J=1,7)=SOURCE PULSE DURATIONS(USEC)
C       YOU CAN INPUT AS MANY AS SIX DIFFERENT PULSE LENGTHS.  THE LAST
C       ENTRY MUST BE BLANK(ZERO).  THE SUBROUTINE CONVOLUTES THE MONTE
C       CARLO PROBABILITY PER UNIT TIME DATA WITH EACH OF THESE SQUARE
C       PULSES, AND, FOR EACH PULSE, WRITES THE TIME-DEPENDENT POWER TO
C       THE DETECTOR, FOR UNIT SOURCE PULSE POWER, FOR EACH SIGNIFICANT
C       ORDER OF MULTIPLE SCATTERING, AND FOR THE TOTAL OF ALL ORDERS.
C-----------------------------------------------------------------------
```

```
C      CARD IDENTIFIER : GO
C      VARIABLES READ : NONE
C       RUN TERMINATION CARD (MUST BE LAST CARD READ).
C-----------------------------------------------------------------------
       IERR=0
C
C***   THE FOLLOWING STATEMENT MAY NEED TO BE CHANGED OR ELIMINATED
C***   FOR USE ON NON-HP1000 COMPUTERS.
C
       SEED0=735.34829
       NDIM=100
       ND1=NDIM-1
       DTAUM=0.2
       GAMMA=EXCO
       SINGWV=WAVE
C
C** INPUT DATA INITIALIZATIONS.
C
       IF(IZERO.NE.0) GO TO 600
       N1=0
       N2=0
       ITIME=0
       ISO=0
       DO 599 LL=1,3
       XS(LL)=0.
       XD(LL)=0.
       A(LL)=0.
       TPU(LL)=0.
   599 TPU(LL+3)=0.
       TPU(7)=0.
       THES=0.
       PHIS=0.
       ASMM=0.
       THED=0.
       PHID=0.
       THEV=0.
       ACM=0.
       THE=0.
       PHE=0.
       PSE=0.
       ZG=0.
       ALBG=0.
       LLMAX=65
       IZERO=1
   600 CONTINUE
C
C***   READ BASCAT DATA SET RECORDS UNDER CARD-INDEPENDENT FORMAT
C
       DO 700 K=1,7
       READ(IOIN,610) IA,IA1,(DAT(I),I=1,7)
   610 FORMAT(2A2,1X,7(E9.4,1X))
       DO 615 JJ=1,8
       IF(IA.NE.IAL(JJ)) GO TO 615
       INOPT=JJ
       IF(INOPT.EQ.7) GO TO 701
       GO TO 620
   615 CONTINUE
       IF((K.EQ.7).AND.(JJ.EQ.8)) GO TO 697
       GO TO 695
   620 CONTINUE
       GO TO (621,622,623,624,625,626),INOPT
   621 N1=IFIX(DAT(1))
       N2=IFIX(DAT(2))
       ITIME=IFIX(DAT(3))
       GO TO 700
   622 XS(1)=DAT(1)
       XS(2)=DAT(2)
       XS(3)=DAT(3)
       THES=DAT(4)
       PHIS=DAT(5)
```

```
      ASMM=DAT(6)
      GO TO 700
  623 XD(1)=DAT(1)
      XD(2)=DAT(2)
      XD(3)=DAT(3)
      THED=DAT(4)
      PHID=DAT(5)
      THEV=DAT(6)
      ACM=DAT(7)
      GO TO 700
  624 A(1)=DAT(1)
      A(2)=DAT(2)
      A(3)=DAT(3)
      THE=DAT(4)
      PHE=DAT(5)
      PSE=DAT(6)
      ISO=IFIX(DAT(7))
      GO TO 700
  625 ZG=DAT(1)
      ALBG=DAT(2)
      GO TO 700
  626 CONTINUE
      DO 627 NN=1,7
  627 TPU(NN)=DAT(NN)
      GO TO 700
C
C***  ERROR RETURNS
C
  695 CONTINUE
      WRITE(IOOUT,696)
  696 FORMAT(1H0,20X,89H***BASCAT ERROR*** INPUT CARD DETECTED WHICH DOE
     +S NOT MATCH ANY CORRECT INPUT IDENTIFIERS /)
      IERR=1
      GO TO 777
  697 CONTINUE
      WRITE(IOOUT,698)
  698 FORMAT(1H0,20X,66H***BASCAT ERROR*** TOO MANY INPUT CARDS OR GO SE
     +NTINEL NOT PRESENT /)
  700 CONTINUE
  701 CONTINUE
C
C***  GEOMETRICAL OPTION DATA TRANSFER
C
      IF(IGEOSW.NE.1) GO TO 111
      DO 110 I=1,3
      XS(I)=PTS(I+6)-PTS(I+12)
  110 XD(I)=PTS(I+3)-PTS(I+12)
  111 CONTINUE
C
C***  GENERATE INTERPOLATED, RENORMALIZED PHASE FUNCTION
C
      CALL PFUNC(ISO)
      IF(GAMMA.EQ.0.0) GAMMA=BE(1)
      ALBEDO=ALBED(1)
      REWIND IPHFUN
      ALB(1)=ALBEDO
      ALB(2)=ALBG
C
C***  DETERMINE POWER OF 2 (KMAX) CORRESPONDING TO NUMBER OF PHASE
C***  FUNCTION VALUES PRESENT.
C
      LMAX=LLMAX
      LMM1=LMAX-1
      KMAX=IFIX(ALOG(FLOAT(LMM1))/0.693147)
C
C***  CC = SPEED OF LIGHT (KM/MICROSECOND)
C
      CC=0.3
      THV=PIRAD*THEV
      IF(THV.LE.1.E-30) THV=0.
```

```
      UV=COS(THV)
C
C***  SET RANDOM NUMBER SEED
C
      SEED=(2*ITIME-1)*SEEDO
      AKM=ACM*1.E-5
      AKSQ=AKM**2
C
C***  SET MINIMUM DISTANCE ALIM (AND ITS SQUARE BLIM) SEPARATING
C***  THE DETECTOR DISK AND A PHOTON SCATTERING POINT.
C
      ALIM=AKM*10.
      BLIM=ALIM**2
      FAC=THV*AKSQ/8.
      BT=.61*WAVE*1.E-3
      THSP=0.
C
C***  DETERMINE DIFFRACTION-LIMITED SOURCE BEAMSPREAD
C
      IF(ASMM.GT.0.)THSP=BT/ASMM
C
C***  GENERATE ROTATION MATRIX RE(  ) FOR CONVERSION FROM STANDARD
C***  FRAME OF REFERENCE TO CLOUD FRAME OF REFERENCE
C
      CALL MATRX(THE,PHE,SS,R)
      AT=PIRAD*PSE
      IF(AT.LE.1.E-30) AT=0.
      DO 1 I=1,3
      DO 1 J=1,3
    1 RS(I,J)=0.
      RS(1,1)=COS(AT)
      RS(1,2)=SIN(AT)
      RS(2,1)=-RS(1,2)
      RS(2,2)=RS(1,1)
      RS(3,3)=1.
      DO 2 I=1,3
      DO 2 J=1,3
      RE(I,J)=0.
      DO 2 K=1,3
    2 RE(I,J)=RE(I,J)+RS(I,K)*R(K,J)
C
C***  GENERATE ROTATION MATRIX RS(  ) FOR CONVERSION FROM SOURCE
C***  CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE
C
      CALL MATRX(THES,PHIS,SS,RS)
C
C***  ECHO INPUT PARAMETERS
C
      WRITE (IOOUT,6800)
      WRITE (IOOUT,4800)
      WRITE (IOOUT,4900)
      WRITE (IOOUT,5000)
      WRITE (IOOUT,4800)
      WRITE (IOOUT,4700)
      WRITE (IOOUT,5100)
      IF (ISO.EQ.0) WRITE (IOOUT,5200)
      IF (ISO.EQ.1) WRITE (IOOUT,5201)
      IF (ISO.EQ.2) WRITE (IOOUT,5202)
      IF (ISO.EQ.3) WRITE (IOOUT,5203)
      IF (ISO.EQ.4) WRITE (IOOUT,5204)
      IF (ISO.EQ.5) WRITE (IOOUT,5205)
      IF (ISO.EQ.6) WRITE (IOOUT,5206)
      IF (ISO.EQ.7) WRITE (IOOUT,5207)
      IF (ISO.EQ.8) WRITE (IOOUT,5208)
      IF (ISO.EQ.9) WRITE (IOOUT,5209)
      IF (ISO.EQ.10) WRITE (IOOUT,5210)
      IF (ISO.EQ.11) WRITE (IOOUT,5211)
      IF (ISO.EQ.12) WRITE (IOOUT,5212)
      WRITE (IOOUT,5600) WAVE,ALBEDO
      WRITE (IOOUT,5700) GAMMA
```

```
      WRITE(IOOUT,5601)                                                  BAS04210
      WRITE(IOOUT,5602)                                                  BAS04220
      WRITE(IOOUT,5603)                                                  BAS04230
      WRITE(IOOUT,5701)                                                  BAS04240
      WRITE(IOOUT,5702)(XS(K),K=1,3)                                     BAS04250
      WRITE(IOOUT,5704)THES                                              BAS04260
      WRITE(IOOUT,5705)PHIS                                              BAS04270
      WRITE(IOOUT,5703)ASMM                                              BAS04280
      WRITE(IOOUT,5706)THSP                                              BAS04290
      WRITE (IOOUT,5900)                                                 BAS04300
      WRITE (IOOUT,6000) THEV                                            BAS04310
      WRITE (IOOUT,6100) ACM                                             BAS04320
      WRITE(IOOUT,6201)(XD(K),K=1,3)                                     BAS04330
      WRITE (IOOUT,6400) THED                                            BAS04340
      WRITE (IOOUT,6500) PHID                                            BAS04350
      WRITE(IOOUT,6501)                                                  BAS04360
      WRITE(IOOUT,6502)ZG,ALBG                                           BAS04370
      DO 10 K=1,3                                                        BAS04380
   10 ASQ(K)=A(K)**2                                                     BAS04390
      DO 12 K=1,3                                                        BAS04400
   12 Y(K)=A(K)*2.*GAMMA                                                 BAS04410
C                                                                        BAS04420
C***  DETERMINE LARGEST OPTICAL DEPTH PRESENT IN AEROSOL CLOUD           BAS04430
C                                                                        BAS04440
      CALL GMAX(3,TAU)                                                   BAS04450
C                                                                        BAS04460
C***  SET TIME AND DISTANCE INCREMENTS AND LIMITS                        BAS04470
C                                                                        BAS04480
      NTMAX=50                                                           BAS04490
      D=TAU/GAMMA                                                        BAS04500
      DELD=5.*D/NTMAX                                                    BAS04510
      DELT=DELD/CC                                                       BAS04520
      DMAX=5.5*D                                                         BAS04530
C                                                                        BAS04540
C***  BEGIN CLOUD SUBBLOCK                                               BAS04550
C                                                                        BAS04560
C######################################################################## BAS04570
C                                                                        BAS04580
C     THE FOLLOWING BLOCK OF WRITE STATEMENTS MUST BE UNCOMMENTED        BAS04590
C     IN ORDER TO OUTPUT DATA TO A USER-DEFINED PLOT FILE (NPLOTU).      BAS04600
C                                                                        BAS04610
C     THE OUTPUT QUANTITIES IN THIS BLOCK ARE THE FOLLOWING :            BAS04620
C       WAVE = WAVELENGTH (MICROMETERS)                                  BAS04630
C       ISO  = AEROSOL TYPE (VALID RANGE, 0-12)                          BAS04640
C       TAU  = OPTICAL DEPTH ALONG LONGEST AXIS OF CLOUD ELLIPSOID       BAS04650
C       N1   = NUMBER OF PARTIAL RUNS WITHIN THIS BASCAT RUN             BAS04660
C       XS(  )= SOURCE XYZ POSITION ARRAY (KILOMETERS)                   BAS04670
C       THES = SOURCE VECTOR POLAR ANGLE (DEGREES)                       BAS04680
C       PHIS = SOURCE VECTOR AZIMUTHAL ANGLE (DEGREES)                   BAS04690
C       ASMM = SOURCE APERTURE RADIUS (MILLIMETERS)                      BAS04700
C       THSP = HALF-ANGLE OF SOURCE DIFFRACTION CONE (RADIANS)           BAS04710
C       XD(  )= DETECTOR XYZ POSITION ARRAY (KILOMETERS)                 BAS04720
C       THED = DETECTOR VECTOR POLAR ANGLE (DEGREES)                     BAS04730
C       PHID = DETECTOR VECTOR AZIMUTHAL ANGLE (DEGREES)                 BAS04740
C       ACM  = DETECTOR APERTURE RADIUS (CENTIMETERS)                    BAS04750
C       THEV = HALF-ANGLE OF DETECTOR FIELD OF VIEW (DEGREES)            BAS04760
C       A(  ) = CLOUD ELLIPSOID PRINCIPAL HALF-AXIS ARRAY (KILOMETERS)   BAS04770
C       ALB(1)= SINGLE-SCATTERING ALBEDO OF CLOUD AEROSOL                BAS04780
C       ALB(2)= ALBEDO OF GROUND PLANE                                   BAS04790
C                                                                        BAS04800
C     WRITE(NPLOTU,9111) WAVE,ISO,TAU,N1                                 BAS04810
C9111 FORMAT(E9.4,1X,I2,1X,E9.4,1X,2(I2,1X))                             BAS04820
C     WRITE(NPLOTU,9222)XS(1),XS(2),XS(3),THES,PHIS,ASMM,THSP            BAS04830
C     WRITE(NPLOTU,9222)XD(1),XD(2),XD(3),THED,PHID,ACM,THEV             BAS04840
C9222 FORMAT(12(E9.4,1X))                                                BAS04850
C     WRITE(NPLOTU,9222) A(1),A(2),A(3),ALB(1),ALB(2)                    BAS04860
C######################################################################## BAS04870
      WRITE (IOOUT,6900)                                                 BAS04880
      WRITE (IOOUT,7000)(A(K),K=1,3)                                     BAS04890
      WRITE (IOOUT,7100)THE,PHE,PSE                                      BAS04900
```

```
      WRITE (IOOUT,7200)(Y(K),K=1,3)
C
C***  GENERATE ROTATION MATRIX R(  ) FOR CONVERSION FROM DETECTOR
C***  CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE
C
      CALL MATRX(THED,PHID,SD,R)
C
C***  SET LIMIT NSCAM ON HIGHEST SIGNIFICANT SCATTERING ORDER (NSCAM-1)
C
      NSCAM=3.0*ALBEDO*TAU+2.0
      IF (NSCAM.GT.10) NSCAM=10
      IF(NSCAM.LT.3)NSCAM=3
      NSCA1=NSCAM-1
      AT=0.
      DO 4 K=1,3
    4 AT=AT+RE(K,3)**2/ASQ(K)
      BT=1.-AT*ZG
      IF(BT.LE.0.)GO TO 4002
      AT=ZG+D
      IF(AT.GT.0.)GO TO 3
      ZG=-D
      WRITE(IOOUT,9004)ZG
    3 CONTINUE
      IF(XS(3).GE.ZG)GO TO 16
      XS(3)=ZG
      WRITE(IOOUT,9003)XS(3)
   16 CONTINUE
C
C***  DETERMINE DISTANCES ELD(1), ELD(2) FROM DETECTOR TO NEAREST AND
C***  FARTHEST CLOUD BOUNDARIES ALONG DETECTOR AXIS
C
      CALL ELM(XD,SD,ELD)
C
C***  DETERMINE DISTANCES EL(1), EL(2) FROM SOURCE TO NEAREST AND
C***  FARTHEST CLOUD BOUNDARIES ALONG SOURCE AXIS
C
      CALL ELM(XS,SS,EL)
C
C***  PERFORM GEOMETRICAL ERROR CHECKS
C
      IF((EL(2).LE.0.).AND.(ELD(2).LE.0.))GO TO 4003
      IF((EL(2).GT.0.).AND.(ELD(2).GT.0.))GO TO 6
      IF(EL(2).LE.0.)GO TO 7
      IF(ALB(2).LE.0.) GO TO 4004
      IF(SD(3).GE.0.)GO TO 4001
      AT=EL(1)
      CT=(ZG-XD(3))/SD(3)
      GO TO 9
    7 CONTINUE
      IF(ALB(2).LE.0.) GO TO 4005
      IF(SS(3).GT.0.)GO TO 4000
      AT=(ZG-XS(3))/SS(3)
      GO TO 8
    6 AT=EL(1)
C
C***  DETERMINE MINIMUM POSSIBLE TRAVERSE DISTANCE ELMIN
C
    8 CT=ELD(1)
    9 BT=0.
      DO 11 K=1,3
      X(K)=XS(K)+AT*SS(K)
      Y(K)=XD(K)+CT*SD(K)
   11 BT=BT+(X(K)-Y(K))**2
      IF(BT.LE.1.E-30) BT=0.
      ELMIN=SQRT(BT)+AT+CT
      IF(ELMIN.LT.ALIM) ELMIN=ALIM
C
C***  DETERMINE VIRTUAL SOURCE POINT XV(  )
C
      DSA=0.
```

```
      IF(ASMM.GT.0.)DSA=ASMM*1.E-6/TAN(THSP)
      DO 305 K=1,3
  305 XV(K)=XS(K)-DSA*SS(K)
C
C***  DETERMINE WHETHER NEAR-CLOUD LIDAR BIASING IS NEEDED.
C
      DOTSD=0.
      DOTXD=0.
      DO 47 KL=1,3
      DOTXD=DOTXD+(XD(KL)-XS(KL))**2
   47 DOTSD=DOTSD+SS(KL)*SD(KL)
      IF(DOTXD.GT.BLIM) DOTSD=0.
      REWIND KSTOR
C
C***  START PHOTON LOOPS
C
      DO 1000 I1=1,N1
C
C***  CHECK TO SEE IF THIS IS THE FIRST RUN WITH NO PRIOR RESULTS USED
C
      IF((I1.EQ.1).AND.(ITIME.EQ.1))GO TO 1303
C
C***  RELOAD WITH RESULTS FROM PREVIOUS PARTIAL RUN
C
      DO 1301 NSCA=1,NSCAM
      READ(KSTOR)ENC(NSCA)
      DO 1301 II=1,NTMAX
      READ(KSTOR)EN(NSCA,II)
 1301 CONTINUE
      REWIND KSTOR
      GO TO 1302
 1303 CONTINUE
C
C***  INITIALIZATION FOR FIRST RUN
C
      DO 1300 NSCA=1,NSCAM
      ENC(NSCA)=0.
      DO 1300 NT=1,NTMAX
 1300 EN(NSCA,NT)=0.0
 1302 CONTINUE
C
C***  LOAD EXPECTED CUMULATIVE NUMBER OF PHOTONS
C
      DO 1304 NS=1,NSCAM
 1304 ENC(NS)=ENC(NS)+N2
C
C***  DETERMINE VECTOR FROM VIRTUAL SOURCE POINT TO DETECTOR
C
      DO 555 K=1,3
  555 Y(K)=XD(K)-XV(K)
C
C***  START INNER PHOTON LOOP
C
      DO 900 I2=1,N2
C
C***  INITIALIZE PHOTON DIRECTION AND CALCULATE DIRECT BEAM
C***  CONTRIBUTIONS (IF ANY)
C
      CALL START(SS)
      IF(STH.GT.0.)GO TO 150
      GO TO 900
  150 NSCA=1
C
C***  START MULTIPLE SCATTERING LOOP
C
      I2FLG=0
      ICOND=0
 1700 NSCA=NSCA+1
C
C***  NEAR-CLOUD LIDAR BIASING IS ACTIVE WHEN ICOND=1. 10 PERCENT OF ALL
```

```
C**** FIRST ORDER PHOTONS USE AN ALTERNATE MODE OF BIASING IN THIS
C**** SITUATION. THE REMAINING 90 PERCENT ARE NORMALLY BIASED IN ALL
C**** ORDERS.
C
      IF((NSCA.NE.2).OR.(DOTSD.LE.0.99).OR.(ELD(1).GT.0.01))GO TO 707
      ICOND=1
      IF(FLOAT(I2).LE.(FLOAT(N2)/10.))I2FLG=1
  707 CONTINUE
C
C***  MOVE PHOTON TO NEXT EVENT POINT VIA BIASED TRAVERSE AND DETERMINE
C***  POWER CONTRIBUTIONS
C
      CALL TRAVRS(JTYPE,I2FLG,ICOND)
      IF((NSCA.EQ.NSCA1).OR.(STH.LE.0.))GO TO 900
C
C***  DETERMINE SCATTERING DIRECTION FOR NEXT TRAVERSE
C
      CALL GAS(JTYPE)
C
C***  DETERMINE BIASING DIRECTIONS FOR NEXT TRAVERSE
C
      CALL ELM(XA,SA,EL)
      IF(EL(2).LE.EL(1))GO TO 900
      GO TO 1700
  900 CONTINUE
C
C***  END MULTIPLE SCATTERING LOOP, BEGIN CONVOLUTION BLOCK
C
C***  WRITE PARTIAL RUN RESULTS INTO STORAGE FILE FOR USE BY NEXT RUN
C
      DO 2201 NSCA=1,NSCAM
      WRITE(KSTOR)ENC(NSCA)
      DO 2201 II=1,NTMAX
      WRITE(KSTOR)EN(NSCA,II)
 2201 CONTINUE
      REWIND KSTOR
C
C***  OUTPUT PARTIAL RUN RESULTS FOR STEADY STATE POWER
C
      WRITE (IOOUT,7400)
      WRITE(IOOUT,7500)
      WRITE(IOOUT,7600)
      PTOT=0.
C######################################################################
C       THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO
C       A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.
C
C          NSCAM= 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING
C
C     WRITE(NPLOTU,9333) NSCAM
C######################################################################
      DO 2110 NS=1,NSCAM
      PE=0.
      NS1=NS-1
      DO 2115 NT=1,NTMAX
 2115 PE=PE+EN(NS,NT)
      PE=PE/ENC(NS)
      PTOT=PTOT+PE
C######################################################################
C       THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO
C       A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.
C
C          NS1   = ORDER OF SCATTERING
C          PE    = OBSERVED STEADY STATE POWER FOR THIS ORDER
C          ENC( )= TOTAL NUMBER OF PHOTONS COUNTED FOR THIS ORDER
C     WRITE(NPLOTU,9333) NS1,PE,ENC(NS)
C######################################################################
      WRITE(IOOUT,7700)NS1,PE,ENC(NS)
 2110 CONTINUE
C######################################################################
```

```
C       THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO
C       A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.
C
C         PTOT = OBSERVED TOTAL STEADY STATE POWER (ALL ORDERS)
C
C9333 FORMAT(I2,1X,2(E9.4,1X))
C     WRITE(NPLOTU,    9222) PTOT
C**********************************************************************
      WRITE(IOOUT,7701)PTOT
      DO 2500 NSCA=1,NSCAM
C
C***  DETERMINE INDEX OF TIME OF LAST NONZERO POWER VALUE FOR EACH
C***  ORDER
C
      CALL SMOOZ(NSCA,NO)
      NM(NSCA)=1+NO
      IOR(NSCA)=NSCA-1
 2500 Y(NSCA)=NM(NSCA)
C
C***  DETERMINE LATEST TIME INDEX OF NONZERO POWER FOR ALL ORDERS
C
      CALL GMAX(NSCAM,YMAX)
      NMA=YMAX
      JP=1
 2900 JP=JP+1
      AT=TPU(JP)
      IF (AT.LE.0.0) GO TO 3000
      GO TO 2900
 3000 JPMAX=JP-1
      WRITE (IOOUT,7400)
      WRITE (IOOUT,7900) JPMAX
C
C***  BEGIN PULSE LOOP
C
      DO 3800 JP=1,JPMAX
      WRITE (IOOUT,7400)
      TP=TPU(JP)
      AT=(NMA+4)*DELT
      IF(TP.GT.AT)TP=AT
      IF(TP.LE.DELT)TP=DELT+1.E-3
      NP=1.001+TP/DELT
      NMAX=NMA+NP
      IF(NMAX.GT.NDIM)NMAX=NDIM
      NP=NMAX-NMA
      TP=(NP-1)*DELT
      TMAX=(NMAX-1)*DELT
      WRITE (IOOUT,8100) JP,TP
      WRITE (IOOUT,8200) JP,TMAX
      IF(TP.LE.0.)GO TO 3999
      DO 3400 NSCA=1,NSCAM
      NMS=NM(NSCA)
      NMS1=NMS-1
C
C***  NORMALIZE RETURN POWER BY DIVIDING CUMULATIVE POWER BY CUMULATIVE
C***  NUMBER OF PHOTONS
C
      DO 3200 N=1,NMS1
 3200 EN(NSCA,N)=EN(NSCA,N)/ENC(NSCA)
C
C***  PERFORM SQUARE SOURCE PULSE CONVOLUTION WITH PROBABILITIES
C***  PER UNIT TIME
C
      CALL CONV(NP,NMS,NMAX,NSCA)
 3400 CONTINUE
      DO 3500 N=1,NMAX
      Y(N)=0.0
      DO 3500 NSCA=1,NSCAM
 3500 Y(N)=Y(N)+EN(NSCA,N)
      XN=-DELT
      DO 3600 N=1,NMAX
```

```
      XN=XN+DELT
 3600 X(N)=XN
C
C***  OUTPUT TIME-DEPENDENT RESULTS
C
      WRITE (IOOUT,8300)
      WRITE (IOOUT,8400)
      WRITE(IOOUT,8500)(IOR(NS),NS=1,NSCAM)
      WRITE(IOOUT,8501)
C#####################################################################
C       THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO
C        A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.
C
C           NMAX  = NUMBER OF TIME BOXES USED FOR TIME-DEPENDENT DATA
C           JP    = INDEX OF INPUT PULSE
C           NSCAM = 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING
C           X( )  = ARRAY OF TIME VALUES FOR EACH TIME BOX
C           Y( )  = ARRAY OF OBSERVED TIME-DEPENDENT TOTAL POWER
C           EN(L,)= ARRAY OF OBSERVED TIME-DEPENDENT POWER FOR ORDER L+1
C
C     IF(I1.LT.N1) GO TO 9777
C     WRITE(NPLOTU,9444) NMAX,JP,NSCAM
C9444 FORMAT(3(I3,1X))
C     WRITE(NPLOTU,9222)(X(LLL),LLL=1,NMAX)
C     WRITE(NPLOTU,9222)(Y(LLL),LLL=1,NMAX)
C     DO 9666 LLL=1,NSCAM
C     WRITE(NPLOTU,9222)(EN(LLL,LLX),LLX=1,NMAX)
C9666 CONTINUE
C9777 CONTINUE
C#####################################################################
      DO 3700 N=1,NMAX
      WRITE (IOOUT,8600) X(N),Y(N),(EN(NSCA,N),NSCA=1,NSCAM)
 3700 CONTINUE
C
C***  RELOAD PARTIAL RUN RESULTS FOR CONVOLUTION WITH NEXT PULSE
C
      DO 3701 NS=1,NSCAM
      READ(KSTOR)ENC(NS)
      DO 3701 II=1,NTMAX
      READ(KSTOR)EN(NS,II)
 3701 CONTINUE
      REWIND KSTOR
 3800 CONTINUE
C
C***  END PULSE LOOP
C
 1000 CONTINUE
C
C***  END OUTER PHOTON LOOP?
C
  777 RETURN
C
C***  ERROR RETURN MESSAGES
C
 3999 CONTINUE
      WRITE(IOOUT,8700)NTMAX
      RETURN
 4000 CONTINUE
      WRITE(IOOUT,8800)
      RETURN
 4001 CONTINUE
      WRITE(IOOUT,9000)
      RETURN
 4002 CONTINUE
      WRITE(IOOUT,9001)
      RETURN
 4003 CONTINUE
      WRITE(IOOUT,9002)
      RETURN
 4004 CONTINUE
```

```
      WRITE(IOOUT,9005)
      RETURN
 4005 CONTINUE
      WRITE(IOOUT,9006)
      RETURN
C*****FORMATS
 4200 FORMAT (7F10.5)
 4400 FORMAT (7I10)
 4500 FORMAT (I10,2F10.5,I10)
 4600 FORMAT (5E15.10)
 4700 FORMAT (1H ,34X,51(1H*))
 4800 FORMAT (1H ,34X,1H*,49X,1H*)
 4900 FORMAT (1H ,34X,1H*,9X,31HMONTE CARLO MULTIPLE SCATTERING,9X,1H*)
 5000 FORMAT (1H ,34X,1H*,16X,18HAEROSOL SCATTERING,15X,1H*)
 5100 FORMAT (1H0,48X,23HPARAMETERS FOR THIS RUN)
 5200 FORMAT(1H0,46X,28HUSER SUPPLIED PHASE FUNCTION)
 5201 FORMAT(1H0,43X,34HMARITIME ARCTIC, VIS=0.1 TO 2.0 KM)
 5202 FORMAT(1H0,47X,26HMARITIME POLAR, VIS=0.2 KM)
 5203 FORMAT(1H0,47X,26HMARITIME POLAR, VIS=2.0 KM)
 5204 FORMAT(1H0,42X,36HCONTINENTAL POLAR, VIS=0.2 TO 2.5 KM)
 5205 FORMAT(1H0,52X,16HWHITE PHOSPHORUS)
 5206 FORMAT(1H0,52X,16HHEXACHLOROETHANE)
 5207 FORMAT(1H0,57X,7HFOG OIL)
 5208 FORMAT(1H0,45X,31HDUST (MODERATE AEROSOL LOADING))
 5209 FORMAT(1H0,46X,28HDUST (HEAVY AEROSOL LOADING))
 5210 FORMAT(1H0,43X,34HMARITIME MODEL B, VIS=5 KM, RH=95%)
 5211 FORMAT(1H0,43X,35HMARITIME MODEL B, VIS=10 KM, RH=90%)
 5212 FORMAT(1H0,43X,35HMARITIME MODEL B, VIS=50 KM, RH=50%)
 5500 FORMAT (1H0,46X,27HUSER SUPPLIED AEROSOL MODEL)
 5600 FORMAT (1H ,36X,11HWAVELENGTH=,F6.3,16H MICROMETERS
     1        7HALBEDO=,F5.3)
 5601 FORMAT(1H0,47X,25HELLIPSOIDAL AEROSOL CLOUD)
 5602 FORMAT(1H ,41X,36HCOORDINATE ORIGIN AT CENTER OF CLOUD)
 5603 FORMAT(1H ,38X,42HZ-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH)
 5700 FORMAT (1H ,36X,31HAEROSOL EXTINCTION COEFFICIENT=,
     1        E10.4,7H KM**-1)
 5701 FORMAT(1H0,51X,17HSOURCE PARAMETERS)
 5702 FORMAT(1H ,36X,27HSOURCE XYZ COORDINATES(KM)=,3(F8.4,1X))
 5703 FORMAT(1H ,36X,27HSOURCE APERTURE RADIUS(MM)=,F7.3)
 5704 FORMAT(1H ,36X,26HSOURCE AXIS POLAR ANGLE  =,F7.3,8H DEGREES)
 5705 FORMAT(1H ,36X,26HSOURCE AXIS AZIMUTH ANGLE=,F7.3,8H DEGREES)
 5706 FORMAT(1H ,36X,26HSOURCE BEAM SPREAD ANGLE =,E10.4,8H RADIANS)
 5900 FORMAT (1H0,50X,19HDETECTOR PARAMETERS)
 6000 FORMAT (1H ,35X,29H CONE OF VIEW HALF-ANGLE     =,F7.3,
     1        8H DEGREES)
 6100 FORMAT (1H ,35X,29H DETECTOR APERTURE RADIUS    =,F7.3,
     1        3H CM)
 6201 FORMAT(1H ,36X,29HDETECTOR XYZ COORDINATES(KM)=,3(F8.4,1X))
 6400 FORMAT (1H ,36X,28HDETECTOR AXIS POLAR ANGLE  =,F7.3,
     1        8H DEGREES)
 6500 FORMAT (1H ,36X,28HDETECTOR AXIS AZIMUTH ANGLE=,F7.3,
     1        8H DEGREES)
 6501 FORMAT(1H0,47X,23HGROUND PLANE PARAMETERS,/,40X,38HISOTROPIC REFLE
     *CTION FROM GROUND PLANE)
 6502 FORMAT(1H ,36X,33HGROUND PLANE Z-COORDINATE ZG(KM)=,F7.3,/,
     *37X,33HGROUND PLANE ALBEDO, ALBG,       =,F7.3)
 6800 FORMAT (1H0,34X,51(1H*))
 6900 FORMAT (1H0,51X,16HCLOUD PARAMETERS)
 7000 FORMAT(1H ,31X,43HELLIPSOID PRINCIPAL XYZ HALF-AXES(KM)      =,
     *3(F8.4,1X))
 7100 FORMAT (1H ,31X,43HEULER ANGLES THE,PHE,PSE OF ELLIPSOID(DEG)=,
     *3(F8.4,1X))
 7200 FORMAT (1H ,31X,43HOPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES   =,
     *3(F8.4,1X))
 7400 FORMAT (1H0,100X)
 7500 FORMAT (1H0,32X,53HSTEADY STATE POWER TO DETECTOR, FOR UNIT SOURCE
     * POWER)
 7600 FORMAT (1H0,37X,5HORDER,3X,18HSTEADY STATE POWER,3X,
     1        17HNUMBER OF PHOTONS)
 7700 FORMAT (1H ,39X,I2,8X,E10.5,8X,E12.6)
```

```
7701 FORMAT (1H ,37X,5HTOTAL,7X,E10.5)
7900 FORMAT (1H0,32X,23HPOWER INTO DETECTOR FOR,I2,6H PULSE
    1          23H(S) OF DIFFERENT LENGTH)
8100 FORMAT (1H0,37X,12HPULSE NUMBER,I2,12H HAS LENGTH ,
    1          E10.4,13H MICROSECONDS)
8200 FORMAT (1H ,23X,34HDETECTOR RESPONSE CUTOFF TIME FOR
    1          12HPULSE NUMBER,I2,4H IS ,E10.4,
    2          13H MICROSECONDS)
8300 FORMAT (1H0,25X,68HDETECTOR RESPONSE, POWER AS A FUNCTION OF TIME,
    * FOR UNIT PULSE POWER)
8400 FORMAT (1H0,55X,21HPOWER FROM EACH ORDER/14X,5HTOTAL)
8500 FORMAT (1H ,3X,4HTIME,6X,5HPOWER,2X,10(5X,I2,4X))
8501 FORMAT(1H ,130(1H-))
8600 FORMAT (12(E10.4,1X))
8700 FORMAT(1H0,6HNTMAX=,I3,80H SHOULD BE DECREASED TO 46. IT IS TOO LA
    *RGE TO ALLOW CONVOLUTION WITH YOUR PULSE)
8800 FORMAT(1H0,97HYOUR INCIDENT PHOTONS NEVER INTERSECT THE CLOUD OR T
    *HE GROUND. CHECK YOUR INPUT SOURCE PARAMETERS)
9000 FORMAT(1H0,72HTHE DETECTOR LOOKS ABOVE THE CLOUD. CHECK YOUR INPUT
    * DETECTOR PARAMETERS)
9001 FORMAT(1H0,65HYOUR GROUND PLANE IS ENTIRELY ABOVE YOUR CLOUD. CHEC
    *K YOUR INPUTS)
9002 FORMAT(1H0,76HNEITHER YOUR SOURCE NOR YOUR DETECTOR LOOK INTO THE
    *CLOUD. CHECK YOUR INPUTS)
9003 FORMAT(1H0,71HYOUR SOURCE WAS UNDERGROUND. IT HAS BEEN PUT AT THE
    *GROUND, WITH XS(3)=,F6.3,2HKM)
9004 FORMAT(1H0,114HYOUR GROUND PLANE WAS TOO FAR AWAY FROM THE CLOUD T
    *O PRODUCE GROUND REFLECTIONS WITHIN THE MAX TIME DELAY ALLOWED.,/,
    *34H THE GROUND PLANE WAS MOVED TO ZG=,F6.3,2HKM)
9005 FORMAT(1H0,91HGROUND PLANE WAS ABSENT AND DETECTOR DOES NOT LOOK A
    *T CLOUD. CHECK YOUR DETECTOR PARAMETERS)
9006 FORMAT(1H0,79HGROUND PLANE WAS ABSENT AND SOURCE DOES NOT ILLUMINA
    *TE CLOUD. CHECK YOUR INPUTS)
     END
```

```
      SUBROUTINE BKWD(JTYPE)
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON/RNDM/ SEED
      COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
     +NTMAX,NSCAM,KMAX,LMAX,LMM1
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
     +BE(16),SINGWV,PF(65),LLMAX
      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
     *ALB(2),ZG,DMAX
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
      COMMON/FGEL/XA(3),D,NSCA
      DIMENSION SBC(3),SBA(3)
      NSCAP=NSCA+1
C
C***  DETERMINE MONTE CARLO BACKWARD TRAVERSE ANGLES THETA (AT)
C***  AND PHI (BT) INSIDE DETECTOR CONE.
C
      BT=TWOPI*RAND  ( SEED)
      AT=THV*RAND  ( SEED)
      IF(AT.LE.1.E-30) AT=0.
      IF(BT.LE.1.E-30) BT=0.
      V=SIN(AT)
      U23=COS(AT)
C
C***  ROTATE BACKWARD TRAVERSE VECTOR INTO STANDARD FRAME OF REFERENCE.
C
      CALL ROTAT(AT,BT,R,SBC)
C
C***  DETERMINE BIASING DISTANCES FOR BACKWARD TRAVERSE.
C
      CALL ELM(XD,SBC,EL)
C
C***  DETERMINE STATISTICAL WEIGHT REX OF BACKWARD TRAVERSE.
C
      REX=1.-EXP(-GAMMA*(EL(2)-EL(1)))
      REXRN=REX*RAND(SEED)
      IF((1.-REXRN).LE.1.E-7) GO TO 15
C
C***  DETERMINE RANDOM DISTANCE FOR BIASED BACKWARD TRAVERSE.
C
      ELBC=-ALOG(1.-REXRN)/GAMMA+EL(1)
      CT=0.
      DO 4 K=1,3
      SBA(K)=XD(K)-XA(K)+SBC(K)*ELBC
    4 CT=CT+SBA(K)**2
      IF(CT.LE.1.E-30) CT=0.
      ELAB=SQRT(CT)
      IF(ELAB.LT.ALIM)GO TO 15
C
C***  DETERMINE TIME BOX INDEX NT FOR THE COMPLETE BACKWARD TRAVERSE.
C
      NT=1.+(DTOT+ELAB+ELBC-ELMIN)/DELD
      IF(NT.LT.0) GO TO 15
      IF(NT.EQ.0) NT=1
      IF(NT.GT.NTMAX)RETURN
      U1=0.
      U2=0.
      DO 5 K=1,3
      SBA(K)=SBA(K)/ELAB
      U1=U1+SA(K)*SBA(K)
    5 U2=U2-SBA(K)*SBC(K)
    6 CALL FIND(U2,PF2)
      IF(JTYPE.EQ.2)GO TO 9
      CALL FIND(U1,PF1)
      GO TO 10
    9 CONTINUE
      CALL ELM(XA,SBA,EL)
      ELAB=ELAB-EL(1)
      PF1=1.
   10 DOM=FAC*ALB(1)*STH*REX*V*U23*PF1*PF2*EXP(-GAMMA*ELAB)/CT
```

```
      EN(NSCAP,NT)=EN(NSCAP,NT)+DOM
      RETURN
   15 ENC(NSCAP)=ENC(NSCAP)-1.
      STH=0.
      RETURN
      END
```

```
      SUBROUTINE CONV(NP,NMS,NMAX,NSCA)
C*****THIS SUBROUTINE CONVOLUTES A GIVEN SEQUENCE OF VALUES EN(NSCA,N),
C     N=1,NMS, WITH THE UNIT SQUARE FUNCTION EXTENDING FROM TIME=0 TO
C*****TIME=(NP-1)*DELT.
      DOUBLE PRECISION DBLE,XDBLE(100)
      COMMON/CONB/X(100),Y(100)
      COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
     +NTMAX,NSCAM,KMAX,LMAX,LMM1
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
     +BE(16),SINGWV,PF(65),LLMAX
      NMS1=NMS+1
      NMA=NMS+NP-1
      XDBLE(1)=0.E-00
      DO 100 N=2,NMS
  100 XDBLE(N)=XDBLE(N-1)+DBLE(EN(NSCA,N-1))
      NP1=NP+1
      IF (NP.LT.NMS) GO TO 600
      DO 200 N=1,NMS
  200 Y(N)=XDBLE(N)
      IF (NP.EQ.NMS) GO TO 400
      DO 300 N=NMS1,NP
  300 Y(N)=XDBLE(NMS)
  400 CONTINUE
      DO 500 N=NP1,NMA
  500 Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)
      GO TO 1000
  600 CONTINUE
      DO 700 N=1,NP
  700 Y(N)=XDBLE(N)
      DO 800 N=NP1,NMS
  800 Y(N)=XDBLE(N)-XDBLE(N-NP+1)
      DO 900 N=NMS1,NMA
  900 Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)
 1000 CONTINUE
      IF (NMA.EQ.NMAX) GO TO 1200
      NMA1=NMA+1
      DO 1100 N=NMA1,NMAX
 1100 Y(N)=0.0
      DO 1150 N=1,NMAX
 1150 EN(NSCA,N)=Y(N)
 1200 RETURN
      END
```

```
      SUBROUTINE ELM(X1,S1,EL)
C
C***  THIS SUBROUTINE DETERMINES THE SMALLEST (EL(1)) AND THE LARGEST
C***  (EL(2)) DISTANCES FROM A POINT X1() TO THE SURFACES OF AN
C***  ELLIPSOID ALONG A LINE OF SIGHT DEFINED BY UNIT VECTOR S1().
C
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
      DIMENSION X1(3),S1(3),EL(2)
      AT=0.
      BT=0.
      CT=-1.
      DO 1 K=1,3
      X=0.
      Y=0.
      DO 2 L=1,3
      X=X+RE(K,L)*X1(L)
    2 Y=Y+RE(K,L)*S1(L)
      AT=AT+Y**2/ASQ(K)
      BT=BT+X*Y/ASQ(K)
    1 CT=CT+X**2/ASQ(K)
      DISC=BT**2-AT*CT
      IF(DISC.LT.1.E-30)GO TO 10
      DISC=SQRT(DISC)
      EL(1)=-(BT+DISC)/AT
      EL(2)=(-BT+DISC)/AT
      IF(CT.LE.0.)EL(1)=0.
      IF((EL(1).LT.0.).OR.(EL(2).LT.0.))GO TO 10
      RETURN
   10 EL(1)=0.
      EL(2)=0.
      RETURN
      END
```

```
      SUBROUTINE FIND(U1,PFU)
      COMMON /M05/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
     +NTMAX,NSCAM,KMAX,LMAX,LMM1
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
     +BE(16),SINGWV,PF(65),LLMAX
      L=1
      LL=LMM1
      DO 1 K=1,KMAX
      LL=LL/2
      L=L+LL
      AT=U1-U(L)
      IF(AT.GT.1.E-7)L=L-LL
    1 CONTINUE
      PFU=PF(L)+(U1-U(L))*(PF(L+1)-PF(L))/(U(L+1)-U(L))
      RETURN
      END
```

```
      SUBROUTINE FWRD(SCA,DOM,ELAC)
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON/FGEL/XA(3),D,NSCA
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
      COMMON/RNDM/ SEED
      DIMENSION SCA(3)
      PSI=TWOPI*RAND  ( SEED)
      TH=PI2
      CALL ROTAT(TH,PSI,R,SCA)
      ANR=RAND(SEED)
      IF(ANR.LE.1.E-30) ANR=0.
      RHO=AKM*SQRT(ANR)
      ELSQ=0.
      DO 1 K=1,3
      SCA(K)=XD(K)-XA(K)+SCA(K)*RHO
    1 ELSQ=ELSQ+SCA(K)**2
      IF(ELSQ.LE.1.E-30) ELSQ=1.E-30
      ELAC=SQRT(ELSQ)
      DO 3 K=1,3
    3 SCA(K)=SCA(K)/ELAC
      DOM=AKSQ/(4.*ELSQ)
      RETURN
      END
```

```
      SUBROUTINE GAS(JTYPE)
C
C***  THIS SUBROUTINE DETERMINES A RANDOMLY-SELECTED SCATTERING
C***  DIRECTION USED IN PHOTON TRAVERSES WITHIN THE ELLIPSOIDAL
C***  AEROSOL CLOUD. JTYPE=1 SIGNIFIES THAT THE SCATTERING EVENT AT
C***  WHICH THIS ANGLE IS SELECTED IS WITHIN THE AEROSOL CLOUD.
C***  JTYPE=2 SIGNIFIES THAT THE SCATTERING EVENT IS ON THE GROUND
C***  PLANE.
C
      COMMON/ALL/AT,BT,CT,BLIM
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
     +NTMAX,NSCAM,KMAX,LMAX,LMM1
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
     +BE(16),SINGWV,PF(65),LLMAX
      COMMON/RNDM/  SEED
      COMMON/FGEL/XA(3),D,NSCA
      AT=RAND  ( SEED)
      IF(AT.LE.1.E-30) AT=0.
      IF(JTYPE.EQ.1)GO TO 1
      SA(3)=1.-AT
      ARG=1.-SA(3)**2
      IF(ARG.LT.1.E-30) ARG=0.
      BT=TWOPI*RAND  ( SEED)
      IF(BT.LE.1.E-30) BT=0.
      V1=SQRT(ARG)
      C1=COS(BT)
      S1=SIN(BT)
      SA(1)=V1*C1
      SA(2)=V1*S1
      RETURN
    1 CALL USCA(AT,BT)
      CT=TWOPI*RAND  ( SEED)
      IF(CT.LE.1.E-30) CT=0.
      ARG=1.-BT**2
      IF(ARG.LT.1.E-30) ARG=0.
      V=SQRT(ARG)
      C=COS(CT)
      S=SIN(CT)
      SA(1)=BT*SA(1)+V*(C*C1*SA(3)-S*S1)
      SA(2)=BT*SA(2)+V*(C*S1*SA(3)+S*C1)
      SA(3)=BT*SA(3)-V*C*V1
      AT=SA(1)**2+SA(2)**2
      IF(AT.LT.1.E-10)GO TO 3
      V1=SQRT(AT)
      C1=SA(1)/V1
      S1=SA(2)/V1
      RETURN
    3 C1=C
      S1=S
      V1=0.
      SA(1)=0.
      SA(2)=0.
      BT=SA(3)
      SA(3)=1.
      IF(BT.LT.0.)SA(3)=-1.
      RETURN
      END
```

```
      SUBROUTINE GMAX(IMAX,YMAX)
C
C***  THIS SUBROUTINE DETERMINES THE MAXIMUM VALUE YMAX OF AN INPUT
C***  ARRAY Y( ) OF DIMENSION IMAX.
C
      COMMON/CONB/X(100),Y(100)
      I=0
      YMAX=Y(1)
    1 I=I+1
      IF(I.EQ.IMAX)RETURN
      T=Y(I+1)-YMAX
      IF(T.GT.0.)YMAX=Y(I+1)
      GO TO 1
      END
```

```
      SUBROUTINE MATRX(TH,PH,S,R)
C
C***  THIS SUBROUTINE GENERATES UNIT VECTOR S( ) AND ROTATION MATRIX
C***  R( ) FOR A SET OF INPUT POLAR ANGLES (TH,PH).
C***  THE ROTATION MATRIX R( ) ROTATES A VECTOR DEFINED RELATIVE TO THE
C***  (TH,PH) DIRECTION INTO THE STANDARD SYSTEM OF COORDINATES.
C***  THE UNIT VECTOR S( ) POINTS IN THE (TH,PH) DIRECTION IN THE
C***  STANDARD SYSTEM OF COORDINATES.
C
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DIMENSION S(3),R(3,3)
      AT=PIRAD*TH
      BT=PIRAD*PH
      IF(AT.LE.1.E-30) AT=0.
      IF(BT.LE.1.E-30) BT=0.
      V1=SIN(AT)
      C1=COS(BT)
      S1=SIN(BT)
      S(1)=V1*C1
      S(2)=V1*S1
      S(3)=COS(AT)
      R(1,1)=C1*S(3)
      R(1,2)=S1*S(3)
      R(1,3)=-V1
      R(2,1)=-S1
      R(2,2)=C1
      R(2,3)=0.
      DO 1 K=1,3
    1 R(3,K)=S(K)
      RETURN
      END
```

```
      SUBROUTINE ROTAT(TH,PH,R,S)
      DIMENSION R(3,3),X(3),S(3)
      IF(TH.LE.1.E-30) TH=0.
      IF(PH.LE.1.E-30) PH=0.
      V=SIN(TH)
      X(1)=V*COS(PH)
      X(2)=V*SIN(PH)
      X(3)=COS(TH)
      DO 1 J=1,3
      S(J)=0.
      DO 1 K=1,3
    1 S(J)=S(J)+R(K,J)*X(K)
      RETURN
      END
```

```
      SUBROUTINE SMOOZ(NSCA,NO)                                         SM000010
C*****THIS SUBROUTINE DETERMINES 'NO', THAT VALUE OF I BEYOND WHICH ALL SM000020
C*****Y(I) IN A SEQUENCE ARE ZERO.                                      SM000030
      COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,    SM000040
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                        SM000050
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),       SM000060
     +BE(16),SINGWV,PF(65),LLMAX                                        SM000070
      I=NTMAX+1                                                         SM000080
  100 I=I-1                                                             SM000090
      IF(I.EQ.0)GO TO 200                                               SM000100
      YI=EN(NSCA,I)                                                     SM000110
      IF (YI.GT.0.0) GO TO 200                                          SM000120
      GO TO 100                                                         SM000130
  200 NO=I                                                              SM000140
      RETURN                                                            SM000150
      END                                                               SM000160
```

```
      SUBROUTINE START(SS)
C
C***  THIS SUBROUTINE INITIALIZES PHOTON LAUNCH DIRECTION AND CALCULATES
C***  DIRECT BEAM (ZEROTH ORDER) CONTRIBUTIONS TO RECEIVED POWER.
C
      COMMON/ALL/AT,BT,CT,BLIM
      COMMON/CONB/X(100),Y(100)
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
      COMMON/HIT/UDS,THSP,RS(3,3),XS(3),DSA,XV(3)
      COMMON/FGEL/XA(3),D,NSCA
      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
     *ALB(2),ZG,DMAX
      COMMON/RNDM/ SEED
      COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
     +NTMAX,NSCAM,KMAX,LMAX,LMM1
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
     +BE(16),SINGWV,PF(65),LLMAX
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DIMENSION SS(3)
C
C***  GENERATE RANDOM THETA (AT) AND PHI (BT) PHOTON LAUNCH ANGLES
C***  CONFINED WITHIN SOURCE CONE.
C
      ANR=RAND(SEED)
      IF(ANR.LE.1.E-30) ANR=0.
      AT=THSP*SQRT(ANR)
      BT=TWOPI*RAND  ( SEED)
C
C***  ROTATE LAUNCH VECTOR SA() INTO STANDARD FRAME OF REFERENCE.
C
      CALL ROTAT(AT,BT,RS,SA)
C
C***  DETERMINE HORIZONTAL COMPONENT CT OF LAUNCH VECTOR AND BRANCH
C***  TO VERTICAL TREATMENT (WHICH INCLUDES GROUND PLANE) IF THIS
C***  COMPONENT IS VERY SMALL.
C
      CT=SA(1)**2+SA(2)**2
      IF(CT.LE.1.E-24)GO TO 1
      V1=SQRT(CT)
      C1=SA(1)/V1
      S1=SA(2)/V1
      GO TO 2
C
C***  DEFINE VERTICAL UNIT VECTOR
C
    1 V1=0.
      CT=SA(3)
      SA(3)=1.
      IF((CT.LT.0.).AND.(ALB(2).GT.0.)) SA(3)=-1.
      C1=1.
      S1=0.
      SA(1)=0.
      SA(2)=0.
C
C***  INITIALIZE TOTAL TRAVERSE DISTANCE DTOT AND STATISTICAL
C***  STRENGTH STH.
C
    2 DTOT=0.
      STH=1.
      BT=0.
      IF(AT.LE.1.E-30) GO TO 11
      DAA=DSA/COS(AT)
      DAD=0.
      DO 3 K=1,3
      XA(K)=XV(K)+DAA*SA(K)
      DAD=DAD+Y(K)**2
    3 BT=BT+Y(K)*SS(K)
      IF(DAD.LE.1.E-30) DAD=0.
      DAD=SQRT(DAD)
C
```

```
C***  DETERMINE DIRECT BEAM ATTENUATION DISTANCE AND BIASING DISTANCES
C***  FOR INITIAL PHOTON TRAVERSE.
C
      CALL ELM(XA,SA,EL)
      IF(BT.LE.ALIM) GO TO 4
      IF(DAD.LE.(1.001*DSA)) GO TO 4
      BT=0.
      CT=0.
      DO 5 K=1,3
      CT=CT+SA(K)*SD(K)
    5 BT=BT+Y(K)*SD(K)
      CTSQ=CT**2
      IF(CTSQ.LE.1.E-30) GO TO 4
      ELSD=ABS(BT/CT)-DAA
      CT=0.
      DO 6 K=1,3
    6 CT=CT+(XA(K)+SA(K)*ELSD-XD(K))**2
      IF(CT.GT.AKSQ) GO TO 4
      IF(EL(2).LE.EL(1)) STH=0.
      IF(ELD(1).LE.0.) EL(2)=ELSD
      IF(DAD.LT.ALIM) GO TO 4
      UDS=-BT/DAD
      IF(UDS.LT.UV)RETURN
      NT=1.+(ELSD-ELMIN)/DELD
      IF(NT.LT.0) GO TO 11
      IF(NT.EQ.0) NT=1
      EN(1,NT)=EN(1,NT)+EXP(-GAMMA*(EL(2)-EL(1)))
      RETURN
    4 CONTINUE
      IF((EL(2)-EL(1)).GT.1.E-20) RETURN
      IF((SA(3).GE.0.).OR.(ALB(2).LE.1.E-20)) STH=0.
      RETURN
   11 STH=0.
      RETURN
      END
```

```
      SUBROUTINE TRAVRS(JTYPE,I2FLG,ICOND)
      COMMON/RNDM/  SEED
      COMMON/ALL/AT,BT,CT,BLIM
      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
     *ALB(2),ZG,DMAX
      COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
     +NTMAX,NSCAM,KMAX,LMAX,LMM1
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
     +BE(16),SINGWV,PF(65),LLMAX
      COMMON/FGEL/XA(3),D,NSCA
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON/CONB/X(100),Y(100)
      DIMENSION SDA(3)
      DGAM=EL(2)-EL(1)
      REXIT=.99999
      RX=.99999
      RXX=.99999
      IF((DGAM.LE.0.).AND.(ALB(2).GT.0.))GO TO 9
      IF(DGAM.LE.1.E-7) GO TO 70
      IF(I2FLG.EQ.1) GO TO 5
      IF((SA(3).GE.0.).OR.(ALB(2).LE.0.))REXIT=1.-EXP(-GAMMA*DGAM)
      AT=EL(1)-(ALOG(1.-REXIT*RAND  ( SEED)))/GAMMA
      GO TO 4
    5 CONTINUE
      IF(ELD(1).GT.ALIM)ALIM=ELD(1)
      RXX=(1./GAMMA)*((1./ALIM)-(1./DGAM))
      DENOM=((1./ALIM)-GAMMA*RXX*RAND(SEED))
      IF(DENOM.LE.1.E-7) GO TO 70
      AT=1./DENOM
      RX=EXP(-(GAMMA*AT))*(AT*GAMMA)**2
    4 BT=XA(3)+SA(3)*AT
      IF((BT.GT.ZG).AND.(AT.LE.EL(2)))GO TO 2
    9 JTYPE=2
      AT=(ZG-XA(3))/SA(3)
      GO TO 1
    2 JTYPE=1
    1 STH=ALB(JTYPE)*REXIT*STH*RXX
      IF(STH.LE.0.)RETURN
      DO 3 K=1,3
    3 XA(K)=XA(K)+AT*SA(K)
      DTOT=DTOT+AT
      IF(NSCA.NE.2) GO TO 50
      AT=0.
      DO 8 K=1,3
      SDA(K)=XD(K)-XA(K)
    8 AT=AT+SDA(K)*SDA(K)
      IF(AT.LT.BLIM)GO TO 15
      AT=SQRT(AT)
      BT=0.
      DO 32 K=1,3
      SDA(K)=SDA(K)/AT
   32 BT=BT-SDA(K)*SD(K)
      IF(BT.LT.UV) GO TO 50
      CALL FWRD(SDA,DOM,AT)
      NT=1.+(DTOT+AT-ELMIN)/DELD
      IF(NT.LT.0) RETURN
      IF(NT.EQ.0) NT=1
      IF((NT.EQ.1).AND.(I2FLG.EQ.0).AND.(ICOND.EQ.1))GO TO 50
      IF(NT.GT.NTMAX)RETURN
      CT=0.
      BT=0.
      DO 7 K=1,3
      BT=BT-SDA(K)*SD(K)
    7 CT=CT+SDA(K)*SA(K)
      PFI=1.
      IF(JTYPE.EQ.1)CALL FIND(CT,PFI)
      CALL ELM(XA,SDA,EL)
      AT=AT-EL(1)
      IF(ELD(1).GT.0.)AT=EL(2)-EL(1)
```

```
      DOM=DOM*PFI*STH*EXP(-GAMMA*AT)*BT*RX                              TRA00710
      IF(I2FLG.NE.1) GO TO 13                                           TRA00720
      IF(NT.EQ.1) DOM=DOM*10.                                           TRA00730
      IF(NT.GT.1) DOM=0.                                                TRA00740
      GO TO 14                                                          TRA00750
   13 CONTINUE                                                          TRA00760
      IF(ICOND.EQ.1) DOM=DOM/0.9                                        TRA00770
   14 EN(NSCA,NT)=EN(NSCA,NT)+DOM                                       TRA00780
      GO TO 50                                                          TRA00790
   15 ENC(2)=ENC(2)-1.                                                  TRA00800
      STH=0.                                                            TRA00810
      RETURN                                                            TRA00820
   50 CONTINUE                                                          TRA00830
      IF(I2FLG.EQ.1) GO TO 60                                           TRA00840
      CALL BKWD(JTYPE)                                                  TRA00850
   60 RETURN                                                            TRA00860
   70 ENC(NSCA)=ENC(NSCA)-1.                                            TRA00870
      STH=0.                                                            TRA00880
      RETURN                                                            TRA00890
      END                                                               TRA00900
```

```
      SUBROUTINE USCA(SC,US)
      COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
     +NTMAX,NSCAM,KMAX,LMAX,LMM1
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
     +BE(16),SINGWV,PF(65),LLMAX
      L=1
      LL=LMM1
      DO 1 K=1,KMAX
      LL=LL/2
      L=L+LL
      IF(SUM(L).GT.SC)L=L-LL
    1 CONTINUE
      US=U(L)+(SC-SUM(L))*(U(L+1)-U(L))/(SUM(L+1)-SUM(L))
      RETURN
      END
```

```
      SUBROUTINE SMOKE(WAVE1,ICLMAT,TRANS,IERR)                         SMK00010
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSMK00020
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,             SMK00030
     1      FOGPRB,WNDVEL,WNDDIR,IPASCT                                 SMK00040
      COMMON /GEOMET/ PTS(15),IGEOSW                                    SMK00050
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     SMK00060
      DIMENSION IR(26),ITTL(11,5),TSUB(11),QDIV(11)                     SMK00070
      EXTERNAL JPASCT                                                   SMK00080
C
C**** NEW PROGRAM OPTIONS ADDED AS REVISIONS 03 & 04
C
C     THIS REVISION TO THE MODEL ALLOWS THE OPTION OF PRODUCING A ONE
C     DIMENSIONAL "SNAPSHOT" OF A SMOKE SCREEN DUE TO ANY NUMBER OF
C     MUNITIONS AT SOME SPECIFIED TIME; OUTPUT FOR THIS OPTION IS
C     CROSSWIND TRANSMISSION AS A FUNCTION OF DOWNWIND DISTANCE AT
C     A SINGLE GIVEN TIME RATHER THAN TRANSMISSION AS A FUNCTION
C     OF TIME AT A SINGLE GIVEN LINE OF SIGHT.
C
C     NEW INPUTS ARE ENTERED THROUGH THE "NAME" CARD AS:
C     NAME
C          STIME = SINGLE GIVEN TIME AT WHICH SCREEN IS TO BE SAMPLED
C          FRONT = LENGTH OF SCREEN TO BE SAMPLED (ALONGWIND)
C          DELX = INCREMENTS BETWEEN CONTIGUOUS LINES OF SIGHT
C          MCUOPT = OPTION TO SUPRESS INTERMEDIATE OUTPUT (1 = SUPRESS)
C
C     ALSO THE TIME OF DETONATION FOR EACH MUNITION IS REQUIRED (IN
C     SECONDS) AS THE FOURTH ENTRY ON THE "MUNC" CARD
C
C**** THE CAVEATS FOR THE NEW OPTION ARE:
C     1) THE LINES OF SIGHT MUST BE CROSSWIND
C     2) STARTING POINT FOR SAMPLING IS THE OBSERVER-TARGET
C        COORDINATES ENTERED ON THE "OBSC" AND "TARC" CARDS
C     3) THE PRNT OPTION ON THE "OUTP" CARD MUST BE ZERO
C     4) A PLOT FILE OPTION HAS BEEN ADDED AS THE FOURTH ENTRY ON
C        THE "OUTP" CARD (NPLT=1 WILL CREATE A PLOT FILE OF THE
C        FINAL RESULTS ON UNIT NPLOTU - SEE EOMAIN)
C                                                                       SMK00090
C NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION        SMK00100
C         AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)          SMK00110
C          TO ALTER PERIOD, CHANGE MAXS AND DIMENSIONS OF SMAS, PVOL AND SMK00120
C          CLTOT.                                                       SMK00130
C                                                                       SMK00140
      COMMON /M05/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9),      SMK00150
     *EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)        SMK00160
      DATA MAXS/960/                                                    SMK00170
      DATA QDIV /0.,.3,.50,.60,.65,.70,.80,1.0,1.,1.,1./                SMK00180
C                                                                       SMK00190
C NOTE: FOR SHORTER OR LONGER BURNS AND OBSCURATIONS, USERS MAY         SMK00200
C     SUBSTITUTE FOR THE COMMON BLOCK AND MAXS PARAMETER AND ALTER      SMK00210
C      DIMENSIONS FOR SMAS, PVOL AND CLTOT.                             SMK00220
C                                                                       SMK00230
C     SMOKE MODEL                                                       SMK00240
C     FORMAL INPUTS:                                                    SMK00250
C            WAVE1 = WAVELENGTH IN MICROMETERS (OR 94. GHZ) FROM ALLOWEDSMK00260
C                    BANDS (SEE EXTC RECORD BELOW).                     SMK00270
C            ICLMAT = CLIMATOLOGY FLAG, IF 1 THEN /CLYMAT/ VALUES OVER- SMK00280
C                    RIDE METR RECORD VALUES.                           SMK00290
C            IGEOSW = GEOMETRY FLAG FROM COMMON GEOMET.  IF SET         SMK00300
C                    TO 1 IN EOMAIN, THEN OBSERVER AND TARGET           SMK00310
C                    COORDINATES ARE PASSED TO SMOKE, AND ANGLE         SMK00320
C                    XNORTH SET TO 90. DEGREES.                         SMK00330
C     FORMAL OUTPUTS:                                                   SMK00340
C            TRANS = TRANSMISSION AT TIME ETO (DEFAULT) OR USER         SMK00350
C                    SPECIFIED TIME (SEE OUTP RECORD BELOW) FOR         SMK00360
C                    WAVELENGTH WAVE1 ALONG THE OBSERVER-TARGET         SMK00370
C                    LINE-OF-SIGHT.                                     SMK00380
C            IERR  = ERROR FLAG.  1 IF ERROR IN SMOKE.                  SMK00390
C     USER RECORDS INPUT:                                               SMK00400
C       EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,         SMK00410
C       FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.            SMK00420
```

```
C       PER FIELD BEGINNING IN COL 11.  THE CARDS ARE NOT ORDER          SMK00430
C       DEPENDENT ALTHOUGH VALUES ON EACH RECORD MUST FOLLOW THE ORDER   SMK00440
C       SHOWN IN COMMENTS BELOW.      FORMAT (2A2,6X,7F10.3)             SMK00450
C                                                                        SMK00460
C  THE FOLLOWING ARE REQUIRED RECORDS FOR AT LEAST ONE INPUT SET.        SMK00470
C                                                                        SMK00480
C      IDENT.   VARS.             DESCRIPTION                            SMK00490
C      ------   --------    ---------------------------------------------SMK00500
C                                                                        SMK00510
C       MUNC    XM,YM,ZM,TM  MUNITION COORDINATES AND EVENT TIME         SMK00520
C               (M,M,M,SEC)
C                                                                        SMK00530
C       BART                 COMPUTATION TIMES AND X-AXIS DEFINITION:    SMK00540
C               STO         OUTPUT STARTING TIME  (SEC. SINCE IGNITION)  SMK00550
C               ETO         ENDING TIME FOR CALCULATION  (SEC.)          SMK00560
C               DTO         TIME INCREMENT FOR OUTPUT TABLES (SEC)       SMK00570
C               ANGLX       ANGLE OF POSITIVE X-AXIS WRT NORTH           SMK00580
C                           (DEG. CLOCKWISE WRT NORTH) ASSUMED 90 DEG. IFSMK00590
C                           IGEOSW IS 1 FROM EOMAIN                      SMK00600
C                                                                        SMK00610
C  OPTIONAL DEPENDING ON PARAMETERS CHOSEN:                              SMK00620
C                                                                        SMK00630
C      IDENT.   VARS.             DESCRIPTION                            SMK00640
C      ------   --------    ---------------------------------------------SMK00650
C       OBSC    XO,YO,ZO    OBSERVER COORDINATES (IGNORED IF IGEOSW      SMK00660
C                           IS 1 FROM EOMAIN). (M,M,M)                   SMK00670
C       TARC    XT,YT,ZT    TARGET COORDINATES (IGNORED IF IGEOSW IS     SMK00680
C                           1 FROM EOMAIN)  (M,M,M)                      SMK00690
C       OUTP              OPTIONAL RECORD TO SELECT AMOUNT OF PRINT OUTPUTSMK00700
C                         AND TO SELECT CRITERIA FOR RETURNED            SMK00710
C                         TRANSMISSION.                                  SMK00720
C               PRNT      IF 0., ALL FULL OUTPUT LISTINGS CREATED.       SMK00730
C                         IF 1. IS ENTERED AT ANY POINT, THEN ALL FURTHER SMK00740
C                         OUTPUT IS SUPPRESSED, EXCEPT FOR THE FINAL     SMK00750
C                         ACCUMULATED EFFECTS LISTING OF TOTAL CL AND    SMK00760
C                         TOTAL TRANSMITTANCE FOR COMBINED MULTIPLE INPUT SMK00770
C                         SETS.  (DEFAULT IS PRNT = 0.)                  SMK00780
C               CRITER    SELECTS CHOICE OF TRANSMISSION RETURNED FROM   SMK00790
C                         SMOKE.                                         SMK00800
C                            0. = RETURN TOTAL TRANSMITTANCE COMPUTED AT SMK00810
C                                 LAST TIME ETO. (DEFAULT CASE)          SMK00820
C                            1. = RETURN THE MINIMUM VALUE OF TOTAL      SMK00830
C                                 TRANSMITTANCE COMPUTED FOR WAVELENGTH  SMK00840
C                                 WAVE1.                                 SMK00850
C                            2. = RETURN VALUE OF TOTAL TRANSMITTANCE FOR SMK00860
C                                 WAVELENGTH WAVE1 COMPUTED AT USER-     SMK00870
C                                 SPECIFIED TIME TIMTRN BELOW.           SMK00880
C               TIMTRN    REQUIRED ONLY IF CRITER IS 2., TIME FOR WHICH  SMK00890
C                         TRANSMITTANCE RETURNED IS COMPUTED.  SHOULD BE SMK00900
C                         CLOSE OR EQUAL TO A TABLE TIME (AS DETERMINED BYSMK00910
C                         THE BART RECORD) FOR ACCURACY.                 SMK00920
C               IPLT      PLOT CODE ADDED IN ORDER TO PLOT OUTPUT ON UNIT
C                         NPLOTU IF IPLT = 1.
C                                                                        SMK00930
C       MUNT              REQUIRED IF BURN RECORD IS NOT USED. OTHERWISE, SMK00940
C                         OPTIONAL.  ANY NON-ZERO VALUES INPUT WILL OVER- SMK00950
C                         RIDE PREVIOUS SOURCE DEFINITIONS (INCLUDING    SMK00960
C                         THOSE FROM THE BURN RECORD.)                   SMK00970
C                                                                        SMK00980
C               XN          NUMBER OF MUNITIONS IGNITED AT THE SAME      SMK00990
C                           LOCATION AND AT THE SAME TIME (DIMENSIONLESS)SMK01000
C               FW          FILL WEIGHT OF ONE MUNITION (LBS.) FOR       SMK01010
C                               WP,PWP,HC OR RP (BUT RATE OF BURN        SMK01020
C                               IN GAL./HR. FOR FOG OIL. NOTE 1 GAL/HR=  SMK01030
C                               0.93 G/S)                                SMK01040
C               TBURN       BURN DURATION FOR THIS MUNITION (SEC)        SMK01050
C               TYPE        TYPE OF SMOKE (DIMENSIONLESS)                SMK01060
C                      1.=WP, 2.=PWP OR WP WICK/WEDGE, 3.=HC,            SMK01070
C                      4.=FOG OIL, 5.=RP                                 SMK01080
C               EFF         MUNITION BURN EFFICIENCY. (PERCENT)          SMK01090
```

```
C                          IF INPUT AS 0., THEN DEFAULT IS USED.
C          YF              YIELD FACTOR (DIMENSIONLESS). IF 0., DEFAULTS
C                          TO RELATIVE HUMIDITY DEPENDENT STRAIGHT
C                             LINE FIT FOR WP,PWP,RP FROM JOHNSON
C                             AND FORNEY. SIMILARLY FOR HC. FOG OIL IS
C                             SET TO 1.
C
C     METR                 REQUIRED IF ICLMAT IS ZERO. OTHERWISE,
C                          NEEDED ONLY FOR E OR F PASQUILL CATEGORY TO
C                          PROVIDE TGRAD WHICH IS NOT AVAILABLE IN
C                          /CLYMAT/.
C
C          RELHUM          RELATIVE HUMIDITY    (PERCENT)
C          UW              WIND VELOCITY (M/SEC)
C          WNDIR           WIND DIRECTION (USUAL MET CONVENTION,
C                          ANGLE IN DEG. CLOCKWISE FROM NORTH OF
C                          DIRECTION FROM WHICH WIND ORIGINATES.)
C          PCAT            PASQUILL CATEGORY (DIMENSIONLESS)
C                             1.-A, 2.-B, 3.-C, 4.-D, 5.-E, 6.-F
C          AIRT            SURFACE AIR TEMPERATURE    (DEG C)
C          TGRAD           VERT TEMP GRADIENT (C DEG/M).
C                          EXAMPLE: TGRAD=(AIRT(10 M)-AIRT(.5 M))/9.5 M
C                                (USED ONLY FOR PASQUILL CATEGORIES E, F)
C
C     EXTC                 OPTIONAL USER OVERRIDE FOR EXTINCTION
C                          COEFFICIENTS.
C                           IF RECORD NOT USED, OR FOR ANY VALUES READ
C                           IN AS 0., THE EXTINCTION COEFF.
C                          DEFAULTS TO ALPHA ARRAY VALUE IN STRANS.
C
C                        INPUT EXTINCTION COEFF. (M**2/G) ORDER ON CARD
C                          CORRESPONDS TO THE BANDS:
C                               0.4-0.7 MICROMETERS
C                               0.7-1.2 MICROMETERS
C                                 1.06  MICROMETERS
C                               3.0-5.0 MICROMETERS
C                               8.0-12. MICROMETERS
C                                 10.6  MICROMETERS
C                                 94.0  GHZ.
C
C     BURN           OPTIONAL -  SELECTS BUILT-IN MUNITION CHARACTER-
C                    ISTICS FROM THE BRATE ROUTINE FOR ONE (XN=1)
C                    MUNITION. VALUES ARE FOR FILL WEIGHT (FW), BURN
C                    DURATION (TBURN), SMOKE TYPE (ITYPE), EFFICIENCY
C                    (EFF).  YIELD FACTOR IS SET TO ZERO SO THAT RH
C                    MODEL DEPENDENT VALUES ARE USED. ANY VALUES READ
C                    IN AS NON-ZERO ON A MUNT RECORD (WHICH IS
C                    OPTIONAL IF A BURN CARD IS USED) WILL OVERRIDE
C                    THE DEFAULTS STORED IN BRATE.
C
C          TYPM            MUNITION TYPE:
C                               0. = USER DEFINED MUNITION SOURCE CHAR.
C                               1. = 155MM HC, M1 CANISTER.
C                               2. = 155MM HC, M2 CANISTER.
C                               3. = 105MM HC CANISTER.
C                               4. = 155MM HC M116B1 PROJ.
C                               5. = 105MM HC M84A1  PROJ.
C                               6. = SMOKE POT HC M5
C                               7. = 60MM  WP M302 CARTRIDGE
C                               8. = 81MM  WP M375A2
C                               9. = 4.2 IN WP M328A1
C                              10. = 155MM WP M110E2
C                              11. = 105MM WP M60A2
C                              12. = 4.2 IN PWP M328A1
C                              13. = 5.   IN PWP ZUNI MK4
C                              14. = 2.75 IN WP WEDGE SUB-MUNITION.
C                              15. = 3.     IN WP WICK  SUB-MUNITION
C                              16. = 6.     IN WP WICK  SUB-MUNITION
C                              17. = 155MM WP WEDGE XM825 (92 SUB-MUN.)
C                              18. = 81MM RP WEDGE NAVY SUB-MUNITION
```

```
C                          19. = 81MM RP WEDGE GERMAN SUB-MUNITION    SMK01800
C                          20. = 155MM RP WEDGE XM803 (228 SUB-MUN.)  SMK01810
C                          21. = FOG OIL GENERATOR (EXAMPLE)          SMK01820
C             BRAT1      BURN-RATE COEFF. (CONST. TERM)               SMK01830
C             BRAT2      BURN-RATE COEFF. (LINEAR TERM)               SMK01840
C             BRAT3      BURN-RATE COEFF. (QUADRATIC TERM)            SMK01850
C             BRAT4      BURN-RATE COEFF. (CUBIC TERM)                SMK01860
C                          RATE COEFF. ARE BASED ON THE SCALED        SMK01870
C                          POLYNOMIAL DESCRIBING THE FRACTIONAL       SMK01880
C                          MASS BURNED AT SCALED TIMES FROM 0.        SMK01890
C                          (IGNITION) TO 1. (BURN-OUT). THE INTEGRAL  SMK01900
C                          OF THE POLYNOMIAL FROM 0. TO 1. MUST       SMK01910
C                          BE 1., IE. THE SCALED TOTAL MASS BURNED    SMK01920
C                          BY TIME 1. IS 100 PERCENT.  THE PROGRAM    SMK01930
C                          WILL CHECK FOR THIS CONDITION AND RE-SCALESMK01940
C                          COEFF. IF NECESSARY.                       SMK01950
C             BRAT5      SPECIAL BURN RATE COEFFICIENT FOR ESTIMATING SMK01960
C                        THE RAPID BURST FOLLOWED BY SLOW BURN OF     SMK01970
C                        SOME MUNITIONS.  THE FUNCTIONAL FORM OF      SMK01980
C                        THE TERM MULTIPLYIED BY BRAT5 IS             SMK01990
C                 .07818 * (358800. / (1. + 358800. T/TBURN))         SMK02000
C                        THOSE MUNITIONS FOR WHICH THE RATE IS NOT    SMK02010
C                        MODELED WILL RECEIVE BRAT1=1., BRAT2,3,4,5 =0SMK02020
C                        (IE., CONSTANT BURN-RATE). THE USER MAY      SMK02030
C                        OVERRIDE THE DEFAULT RATES BY ENTERING VALUESSMK02040
C                        ON THE BURN CARD.                            SMK02050
C         NAME        SCREEN PARAMETERS
C             STIME      TIME AT WHICH SCREEN IS EXAMINED
C             FRONT      SCREEN FRONTAGE TO BE EXAMINED
C             DELX       INCREMENTAL SPACING FOR SCREEN EXAMINATION
C             MCUOPT     OPTION TO SUPRESS INTERMEDIATE OUTPUT WITH
C                        MCU STUDIES OPTION (MCUOPT=1 WILL SUPRESS)
C                                                                     SMK02060
C THE FOLLOWING IDENT RECORDS ARE ALWAYS REQUIRED.                    SMK02070
C                                                                     SMK02080
C         GO          SIGNIFIES TO BEGIN EXECUTION FOR THIS DATA SET. SMK02090
C                     AFTER EXECUTION, ANOTHER SET OF INPUTS MAY BE   SMK02100
C                     READ-IN FOLLOWED BY ANOTHER 'GO' CARD.          SMK02110
C                     ANY VALUES ESTABLISHED FROM PREVIOUS INPUT SETS SMK02120
C                     TO THE ROUTINE ARE STILL IN EFFECT.  THUS DATA  SMK02130
C                     SUCH AS FROM CARD OBSC NEED NOT BE READ AGAIN IFSMK02140
C                     THERE ARE TO BE NO CHANGES TO OBSERVER COORD.,ETC. SMK02150
C                                                                     SMK02160
C         DONE        *** MUST *** BE THE LAST RECORD READ.  DESIGNATES SMK02170
C                     LISTING OF ACCUMULATED EFFECTS IF MORE THAN ONE SMK02180
C                     INPUT SET (DELINEATED BY GO CARDS) WERE INPUT.  SMK02190
C                     ALSO RETURNS CONTOL TO EOSAEL EXEC MODULE.      SMK02200
C                     NOTE THAT A 'DONE' CARD CAN BE USED IN PLACE OF SMK02210
C                     THE FINAL 'GO' CARD IF DESIRED.                 SMK02220
C                                                                     SMK02230
C        OUTPUTS                                                      SMK02240
C            LIST OF INPUT PARAMETERS AND                             SMK02250
C            AT EACH COMPUTATION TIME, TO, THE FOLLOWING:             SMK02260
C                                                                     SMK02270
C             X          CLOUD LENGTH ALONG WIND VECTOR    (M)        SMK02280
C             Z          CLOUD LEADING EDGE HEIGHT ALONG WIND VECTOR  SMK02290
C                          (M)                                        SMK02300
C             YFULL      CLOUD LEADING EDGE FULL-WIDTH PERPENDICULAR  SMK02310
C                          TO WIND VECTOR (M).                        SMK02320
C             PATHL      SMOKE LENGTH OF 1 MUNITION ALONG OBS-TGT LOS SMK02330
C                          (M)                                        SMK02340
C             CL         TOTAL CL OF SMOKE ALONG OBS-TGT LOS (GM/M**2)SMK02350
C                        FOR XN MUNITIONS.                            SMK02360
C             SMTRAN     TRANSMISSION IN SPECTRAL BANDS OF SMOKE ALONGSMK02370
C                          LOS                                        SMK02380
C                                                                     SMK02390
C     SUBROUTINES CALLED...   DIRECTLY: CLSMOK, SCONST, SMASSP, JPASCT,   SMK02400
C            WGGEOM, STRANS, BRATE..  INDIRECTLY: XYZINT, QROOT.      SMK02410
C                                                                     SMK02420
      DATA ITTL/ 2HWH,2HIT,2HE ,2HPH,2HOS,2HPH,2HOR,2HUS,2H (,2HWP,2H) ,SMK02430
```

```
     * 2HPW,2HP ,2HOR,2H W,2HP ,2HWI,2HCK,2H/W,2HED,2HGE,2H  ,
     * 2HHC,2H S,2HMO,2HKE,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,
     * 2HFO,2HG ,2HOI,2HL ,2HSM,2HOK,2HE ,2H  ,2H  ,2H  ,2H  ,
     * 2HRE,2HD ,2HPH,2HOS,2HPH,2HOR,2HUS,2H (,2HRP,2H) ,2H  /
      DATA IR /2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA,
     *2HRC,2HEX,2HTC,2HGO,2H  ,2HBU,2HRN ,2HMI,2HSC,2HOU,2HTP,2HDO,
     *2HNE ,2HNA,2HME /
      DATA BRAT1,BRAT2,BRAT3,BRAT4,BRAT5/1.,0.,0.,0.,1./
      DATA TYPM,XN,FW,TBURN,EFF,YF,RELHUM,UW,WNDIR,ANGLX,PCAT,
     *AIRT,TGRAD /13*0./
      DATA ITGRAD/0/
      BFUN(T)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*BRAT4/4.)))
     *+BRAT5*.07818288*ALOG(1.+358800.*T)
      NUMDIV=9
      NRUNS=0
      TIMTRN=1.
      ICRTR=0
      TRANS=1.
      MCUOPT=0
      NOPRNT=0
      NOMORE=0
      NCY=0
      DO 1 J=1,MAXS
      CLTOT(J)=0.
1     CONTINUE
      DO 2 J=1,8
    2 EXTC(J)=0.
 5    CONTINUE
      NRUNS=NRUNS+1
      MUNRD=0
      KWAVE=0
      IF (WAVE1.GE.0.4.AND.WAVE1.LT.0.7) KWAVE=1
      IF (WAVE1.GE.0.7.AND.WAVE1.LE.1.2) KWAVE=2
      IF (WAVE1.GE.3.0.AND.WAVE1.LE.5.0) KWAVE=4
      IF (WAVE1.GE.8.0.AND.WAVE1.LE.12.0) KWAVE=5
      IF (WAVE1.GT.1.059.AND.WAVE1.LT.1.061) KWAVE=3
      IF (WAVE1.GT.10.59.AND.WAVE1.LT.10.61) KWAVE=6
      IF (WAVE1.GT.93.9.AND.WAVE1.LT.94.1) KWAVE=7
      IF (WAVE1.GT.3188..AND.WAVE1.LT.3195.) KWAVE=7
      IF (KWAVE.EQ.0) GOTO 998
C***  BEGINNING OF READ LOOP
      NCHK=0
      DO 70 I = 1, 15
           IF(I.EQ.15) GO TO 310
      READ(IOIN,20) IR1,IR2,(R1(J),J=1,7)
 20        FORMAT(2A2,6X,7F10.3)
C***  RELATING INPUT DATA TO VARIABLE NAMES.
           IF(IR1.EQ.IR(1).AND.IR2.EQ.IR(2)) GOTO 90
           IF(IR1.EQ.IR(3).AND.IR2.EQ.IR(4)) GOTO 100
           IF(IR1.EQ.IR(5).AND.IR2.EQ.IR(6)) GOTO 110
           IF(IR1.EQ.IR(7).AND.IR2.EQ.IR(8)) GOTO 120
           IF(IR1.EQ.IR(9).AND.IR2.EQ.IR(10)) GOTO 130
           IF(IR1.EQ.IR(11).AND.IR2.EQ.IR(12)) GOTO 140
           IF(IR1.EQ.IR(13).AND.IR2.EQ.IR(14)) GOTO 150
           IF(IR1.EQ.IR(15).AND.IR2.EQ.IR(16)) GOTO 155
           IF(IR1.EQ.IR(17).AND.IR2.EQ.IR(18)) GOTO 105
           IF(IR1.EQ.IR(19).AND.IR2.EQ.IR(20)) GOTO 70
           IF(IR1.EQ.IR(21).AND.IR2.EQ.IR(22)) GOTO 115
           IF(IR1.EQ.IR(23).AND.IR2.EQ.IR(24)) GOTO 154
           IF(IR1.EQ.IR(25).AND.IR2.EQ.IR(26)) GOTO 121
           WRITE(IOOUT,80)
 80        FORMAT(1H ,72HINVALID DATA CARD-DOES NOT CONFORM TO PROPER CO
     *NVENTION IN SMOKE ROUTINE)
      WRITE(IOOUT,30) IR1,IR2,(R1(J),J=1,7)
 30      FORMAT(1H ,2A2,6X,7F10.3)
                                 GO TO 999
90    IF (ICLMAT.EQ.1) GOTO 92
       IF (R1(1).NE.0.)    RELHUM= R1(1)
       IF (R1(2).NE.0.)    UW    = R1(2)
       IF (R1(3).NE.0..OR.R1(2).NE.0.)    WNDIR = R1(3)
```

```
        IF (R1(4).NE.0.)     PCAT  = R1(4)                                      SMK03130
       ICAT=IFIX(PCAT+.0001)                                                    SMK03140
           AIRT  = R1(5)                                                        SMK03150
 92        TGRAD = R1(6)                                                        SMK03160
           ITGRAD=1                                                             SMK03170
           NCHK=1                                                               SMK03180
                               GO TO 70                                         SMK03190
 100       IF (R1(1).NE.0.) XN = R1(1)                                          SMK03200
           IF (R1(2).NE.0.) FW = R1(2)                                          SMK03210
           IF (R1(3).NE.0.) TBURN = R1(3)                                       SMK03220
           IF (R1(4).NE.0.) ITYPE = IFIX(R1(4)+.0001)                           SMK03230
           IF (MUNRD.EQ.0.OR.R1(5).NE.0.) EFF    = R1(5)                        SMK03240
           YF=R1(6)                                                             SMK03250
       MUNRD=1                                                                  SMK03260
           NCHK=1                                                               SMK03270
                               GO TO 70                                         SMK03280
 105   TYPM=R1(1)                                                               SMK03290
       CALL BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,BRAT1,               SMK03300
      *BRAT2,BRAT3,BRAT4,BRAT5)                                                 SMK03310
       IF (IERR.NE.0) WRITE (IOOUT,95) TYPM                                     SMK03320
 95    FORMAT(37H IN SMOKE, ILLEGAL MUNITION TYPE READ ,F5.0)                   SMK03330
       IF (IERR.NE.0) GOTO 999                                                  SMK03340
       IF (R1(2).EQ.0..AND.R1(3).EQ.0..AND.R1(4).EQ.0..AND.R1(5).EQ.0.)         SMK03350
      *                                       GOTO 93                           SMK03360
       BRAT1=R1(2)                                                              SMK03370
       BRAT2=R1(3)                                                              SMK03380
       BRAT3=R1(4)                                                              SMK03390
       BRAT4=R1(5)                                                              SMK03400
       BRAT5=R1(6)                                                              SMK03410
 93    IF (TYPM.GT.0.) MUNRD=1                                                  SMK03420
       NCHK=1                                                                   SMK03430
                           GO TO 70                                             SMK03440
 110       ISTO   = IFIX(R1(1)+.0001)                                           SMK03450
           IETO   = IFIX(R1(2)+.0001)                                           SMK03460
           IDTO   = IFIX(R1(3)+.0001)                                           SMK03470
           ANGLX=R1(4)                                                          SMK03480
       NCHK=1                                                                   SMK03490
                               GO TO 70                                         SMK03500
 115   NOPRNT=0                                                                 SMK03510
       IF (R1(1).NE.0.) NOPRNT=1                                                SMK03520
       CRITER=R1(2)                                                             SMK03530
       ICRTR=IFIX(CRITER+.001)                                                  SMK03540
       IF (ICRTR.GT.2) ICRTR=2                                                  SMK03550
       IF (ICRTR.LT.0) ICRTR=0                                                  SMK03560
       IF (ICRTR.EQ.2) TIMTRN=R1(3)                                             SMK03570
       IPLT=IFIX(R1(4))
       GOTO 70                                                                  SMK03580
 120       XM     = R1(1)                                                       SMK03590
           YM     = R1(2)                                                       SMK03600
           ZM     = R1(3)                                                       SMK03610
           TM     = R1(4)
       NCHK=1                                                                   SMK03620
                               GO TO 70                                         SMK03630
 121       MODE=1
           STIME =R1(1)
           ISRN  =IFIX(STIME+0.0001)
           FRONT =R1(2)
           DELX  =R1(3)
           IF(DELX.LE.0.0) DELX=5.0
           XXX=FRONT/DELX
           NPTS=IFIX(XXX)+1
       MCUOPT=IFIX(R1(4))
                               GO TO 70
 130       XO     = R1(1)                                                       SMK03640
           YO     = R1(2)                                                       SMK03650
           ZO     = R1(3)                                                       SMK03660
       NCHK=1                                                                   SMK03670
                               GO TO 70                                         SMK03680
 140       XT     = R1(1)                                                       SMK03690
           YT     = R1(2)                                                       SMK03700
```

```
      ZT     = R1(3)
      NCHK=1
                                      GO TO 70
 150  DO 152 J=1,7
 152  EXTC(J)=R1(J+1)
      EXTC(8)=0.
   70 CONTINUE
154   NOMORE=1
      IF (NCHK.EQ.0) NRUNS=NRUNS-1
      IF (NCHK.EQ.0) GOTO 980
 155  CONTINUE
C******REDEFINE MUNITION EFFICIENCY IF INPUT AS ZERO
      IF(EFF.GT.0.0)GO TO 11
      IF(ITYPE.EQ.1)EFF=100.
      IF(ITYPE.EQ.2)EFF=65.0
      IF(ITYPE.EQ.3)EFF=40.0
      IF(ITYPE.EQ.4)EFF=100.0
      IF(ITYPE.EQ.5)EFF=50.
   11 CONTINUE
      IF (ICLMAT.NE.1) GOTO 12
      RELHUM=RH
      UW    =WNDVEL
      WNDIR=WNDDIR
      ICAT  =IPASCT
      PCAT=FLOAT(ICAT)
      AIRT  =TEMP
   12 CONTINUE
      IF (IGEOSW.NE.1) GOTO 13
      DISKTM=1000.
C...  CONVERT UNITS FROM KM TO M.
      XT=PTS(1)*DISKTM
      YT=PTS(2)*DISKTM
      ZT=PTS(3)*DISKTM
      XO=PTS(4)*DISKTM
      YO=PTS(5)*DISKTM
      ZO=PTS(6)*DISKTM
      ANGLX=90.
13    CONTINUE
      IF (ITYPE.LT.1.OR.ITYPE.GT.5) IERR=1
      IF (IERR.EQ.1) WRITE (IOOUT,180) ITYPE
180   FORMAT(1X,31HIN SMOKE, INVALID SMOKE TYPE = ,I4)
      IF (IERR.EQ.1) GOTO 999
C***  CHECK BURN RATE FOR 100 PERCENT BURN AT TBURN...
      IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0..AND.BRAT5.EQ.0.)
     *BRAT1=1.
      VNORM=BFUN(1.)
      BRAT1=BRAT1/VNORM
      BRAT2=BRAT2/VNORM
      BRAT3=BRAT3/VNORM
      BRAT4=BRAT4/VNORM
      BRAT5=BRAT5/VNORM
      IF (XN.LE.0.) XN=1.
      IF (ITYPE.EQ.1.AND.TBURN.GT.1.) TBURN=1.
C***  SET UP EXTINCTION COEFF TO BE USED...
      CALL STRANS(CL,SMTRAN,ITYPE,EXTC,0)
      CALL SMASSP(XN,EFF,FW,RELHUM, W,ITYPE,YF)
      TGR=TGRAD
      TGRAD=ABS(TGRAD)
      IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 255
      IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 255
      IF(NRUNS.GT.1) WRITE (IOOUT,172) NRUNS
      IF(NRUNS.EQ.1) WRITE (IOOUT,1720) NRUNS
172   FORMAT(1H1,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,
     *1H*,15X,9HEXECUTION ,I3/51X,1H*,15X,1H*/51X,17(1H*)/)
1720  FORMAT(1H0,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,
     *1H*,15X,9HEXECUTION ,I3/51X,1H*,15X,1H*/51X,17(1H*)/)
C***  REPORTING INPUT DATA.
      IF(ICAT.GE.5.AND.ITYPE.NE.4.AND.ITGRAD.EQ.0)WRITE (IOOUT,98) TGRAD
   98 FORMAT(1X,44HIN SMOKE ROUTINE PASQUILL CATEGORIES E AND F,
     *29H REQUIRE TEMPERATURE GRADIENT/10X,23HIF SMOKE IS EXOTHERMIC.,
```

```
     *20H  A VALUE OF TGRAD = ,F7.2,25H C DEG/M WILL BE ASSUMED./)          SMK04380
      WRITE(IOOUT,170)                                                      SMK04390
 170  FORMAT(5X,15HSMOKE MUNITIONS,22X,25HMETEOROLOGICAL CONDITIONS,25X,SMK04400
     123HEXTINCTION COEFFICIENTS)                                           SMK04410
      WRITE(IOOUT,190) (ITTL(J,ITYPE),J=1,11),UW,EXTC(1)                    SMK04420
 190  FORMAT(3X,11A2,11X,10HWINDSPEED ,14X,F5.1,2X,3HM/S,16X,                SMK04430
     120H0.4-0.7 MICROMETERS ,F7.3,2X,7HM**2/GM)                            SMK04440
      WRITE(IOOUT,200) XN,EXTC(2)                                           SMK04450
 200  FORMAT(3X,10HNO. ROUNDS,1X,F5.0,17X,22HWIND DIRECTION (USUAL ,28X,SMK04460
     120H0.7-1.2 MICROMETERS ,F7.3,2X,7HM**2/GM)                            SMK04470
      IF (ITYPE.NE.4) WRITE(IOOUT,210) FW,WNDIR,EXTC(3)                     SMK04480
 210  FORMAT(3X,11HFILL WEIGHT,F8.3,3H LB,11X,23HMET CONVENTION AZIMUTH)SMK04490
     1,F6.1,2X,7HDEGREES,12X,20H1.06     MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04500
      IF (ITYPE.EQ.4) WRITE(IOOUT,215) FW,WNDIR,EXTC(3)                     SMK04510
 215  FORMAT(3X,9HBURN RATE,2X,F6.1,2X,6HGAL/HR, 8X,23HMET CONVENTION AZSMK04520
     1IMUTH) ,F6.1,2X,7HDEGREES,12X,20H1.06     MICROMETERS ,F7.3,2X,       SMK04530
     27HM**2/GM)                                                            SMK04540
      WRITE(IOOUT,220)TBURN,RELHUM,EXTC(4)                                  SMK04550
 220  FORMAT(3X,9HBURN TIME,F8.1,2X,3HSEC,11X,17HRELATIVE HUMIDITY,7X,      SMK04560
     1F5.1,2X,7HPERCENT,12X,20H3.0-5.0 MICROMETERS ,F7.3,2X,7HM**2/GM)      SMK04570
      LCAT=JPASCT(ICAT)                                                     SMK04580
      WRITE(IOOUT,230) EFF,LCAT,EXTC(5)                                     SMK04590
 230  FORMAT(3X,10HEFFICIENCY,1X,F6.1,2X,7HPERCENT,7X,17HPASQUILL CATEGOSMK04600
     1RY,9X,A1,23X,20H8.0-12. MICROMETERS ,F7.3,2X,7HM**2/GM)               SMK04610
      WRITE (IOOUT,235) YF,AIRT,EXTC(6)                                     SMK04620
 235  FORMAT(3X,12HYIELD FACTOR,F6.2,15X,15HAIR TEMPERATURE,8X,F6.1,2X,     SMK04630
     18HDEGREE C,11X,20H10.6     MICROMETERS ,F7.3,2X,7HM**2/GM)            SMK04640
      WRITE(IOOUT,240)TGRAD,EXTC(7)                                         SMK04650
 240  FORMAT(36X,14HTEMP. GRADIENT,10X,F6.2,9H C DEG./M,11X,                SMK04660
     111H94.0     GHZ,9X,F7.3,2X,7HM**2/GM)                                 SMK04670
      IF (ICAT.GE.5.AND.TGR.LT.0.) WRITE(IOOUT,250)                         SMK04680
 250  FORMAT(36X,24H(ASSUMED POSITIVE INPUT))                               SMK04690
 255  NRAT=0                                                                SMK04700
      IF (BRAT1.NE.0.) NRAT=1                                               SMK04710
      IF (BRAT2.NE.0.) NRAT=2                                               SMK04720
      IF (BRAT3.NE.0.) NRAT=3                                               SMK04730
      IF (BRAT4.NE.0.) NRAT=4                                               SMK04740
      IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 258                              SMK04750
      IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 258
      IF (NRAT.EQ.1) WRITE (IOOUT,175) BRAT1                                SMK04760
      IF (NRAT.EQ.2) WRITE (IOOUT,176) BRAT1,BRAT2                          SMK04770
      IF (NRAT.EQ.3) WRITE (IOOUT,177) BRAT1,BRAT2,BRAT3                    SMK04780
      IF (NRAT.EQ.4) WRITE (IOOUT,178) BRAT1,BRAT2,BRAT3,BRAT4              SMK04790
      IF (NRAT.EQ.0) WRITE (IOOUT,181)                                      SMK04800
 181  FORMAT(40X,19HBURN RATE PROFILE =)                                    SMK04810
      IF (BRAT5.NE.0.) NRAT=5                                               SMK04820
      NPWP=0                                                                SMK04830
      IF ((ITYPE.EQ.2.OR.ITYPE.EQ.5).AND.NRAT.GT.1) NPWP=1                  SMK04840
      IF (NRAT.EQ.5) WRITE (IOOUT,179) BRAT5                                SMK04850
 179  FORMAT(40X,2H+ ,F8.4,41H*0.0781829*(358800./(1.+358800.*T/TBURN)))SMK04860
 175  FORMAT(1H0,40X,19HBURN RATE PROFILE =,F8.4)                           SMK04870
 176  FORMAT(1H0,31X,19HBURN RATE PROFILE =,F8.4,2H +,                      SMK04880
     *F8.4,10H (T/TBURN))                                                   SMK04890
 177  FORMAT(1H0,20X,19HBURN RATE PROFILE =,F8.4,2H +,                      SMK04900
     *F8.4,12H (T/TBURN) +,F8.4,13H (T/TBURN)**2)                           SMK04910
 178  FORMAT(1H0, 9X,19HBURN RATE PROFILE =,F8.4,2H+ ,                      SMK04920
     *F8.4,12H (T/TBURN) +,F8.4,15H (T/TBURN)**2 +,F8.4,                    SMK04930
     *13H (T/TBURN)**3)                                                     SMK04940
C*** PROVIDE COORDINATES...                                                 SMK04950
 258  ANGL=ANGLX+180.-WNDIR                                                 SMK04960
      IF (ANGL.GT.360.) ANGL=ANGL-360.                                      SMK04970
      IF (ANGL.LT.0.   ) ANGL=ANGL+360.                                     SMK04980
      IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 285                              SMK04990
      XP1=XO-XM                                                             SMK05000
      XP2=XT-XM                                                             SMK05010
      YP1=YO-YM                                                             SMK05020
      YP2=YT-YM                                                             SMK05030
      ZPP1=ZO-ZM                                                            SMK05040
      ZPP2=ZT-ZM                                                            SMK05050
      CA=COS(ANGL*PIRAD)                                                    SMK05060
```

```
      SA=SIN(ANGL*PIRAD)
      XPP3=0.
      YPP3=0.
      ZPP3=0.
      XPP1=XP1*CA+YP1*SA
      YPP1=YP1*CA-XP1*SA
      XPP2=XP2*CA+YP2*SA
      YPP2=YP2*CA-XP2*SA
      IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GO TO 285
      WRITE (IOOUT,260)
      WRITE (IOOUT,270) XM,YM,ZM,XPP3,YPP3,ZPP3,XO,YO,ZO,XPP1,YPP1,
     1ZPP1,XT,YT,ZT,XPP2,YPP2,ZPP2
 260  FORMAT(1H0,30X,17HFIELD COORDINATES,20X,43HROTATED COORD.(WIND X-A
     1XIS,MUNITION ORIGIN)/27X,3H(X),8X,3H(Y),8X,3H(Z),20X,4H(XW),
     27X,4H(YW),7X,4H(ZW))
 270  FORMAT(1X,22HMUNITION COORDINATES= ,3(F9.2,2X),6HMETERS,6X,
     13(F9.2,2X),6HMETERS/1X,22HOBSERVER COORDINATES= ,3(F9.2,2X),
     26HMETERS,6X,3(F9.2,2X),6HMETERS/1X,22HTARGET   COORDINATES= ,
     33(F9.2,2X),6HMETERS,6X,3(F9.2,2X),6HMETERS)
      WRITE (IOOUT,275) ANGLX,TM
275   FORMAT(1X,48HANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = ,
     *F7.2,5H DEG.,/,5X,12HEVENT TIME =,F6.1,4H SEC)
      IF(MODE.GT.0) WRITE(IOOUT,1000) MODE,STIME,FRONT,DELX
 1000 FORMAT(5X,5HMODE=,I2,/,5X,12HSCREEN TIME=,F6.1,/,5X,6HFRONT=,F6.1,
     */,5X,11HINCREMENT= ,F6.1)
      IF (NOPRNT.EQ.1.OR.MCUOPT.EQ.1) GO TO 285
      IF(MODE.EQ.0) WRITE(IOOUT,280)
      IF(MODE.GT.0) WRITE(IOOUT,281)
 280  FORMAT(1H0,6X,4HTIME,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,
     110HPATHLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,7X,5H(SEC),3(1X,8H(MET
     2ERS)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMET
     3ERS),/,60X,39H0.4-0.7 0.7-1.2   1.06  3.0-5.0 8.0-12.,
     415H    10.6  94.GHZ,/)
 281  FORMAT(1H0,6X,5H*LOS*,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,10HPAT
     1HLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,4X,8H(METERS),3(1X,8H(METERS
     2)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMETERS
     3),/,60X,40H0.4-0.7  0.7-1.2  1.06  3.0-5.0  8.0-12.,
     415H    10.6  94.GHZ/)
C***  BEGINNING OF CALCULATIONS.
285   IF (W.EQ.0.) GOTO 999
      IF (ICAT.LT.1.OR.ICAT.GT.6) GOTO 999
      IF(IDTO.EQ.0)   IDTO = 1
      IF (ISTO.LE.0) ISTO=IDTO
      IF (IETO.LE.ISTO) IETO=ISTO
C***  CALCULATIONS DEPENDENT ON TIME.
C*** SET COMPUTATION TIME STEP DTIME TO 1 SEC. FOR HC/FOG OIL/PWP/
C    RP AND WP WICKS/WEDGES
C***    BUT TO TABLE REPORT TIME INCREMENT FOR WP
      DTIME=FLOAT(IDTO)
      TRANS=1.0
      TSUB(1)=0.
      TSUB(2)=TBURN
      IF (NPWP.EQ.0) GOTO 620
      IS=IFIX(5.*TBURN+.0001)
      IF (IS.LT.1) GOTO 610
      DO 286 I=2,NUMDIV
      TSUB(I)=1./5.
286   CONTINUE
      DO 600 I=1,IS
      FI=FLOAT(I)/5.
      T=FI/TBURN
      TMS=BFUN(T)
      DO 601 JDIV=2,NUMDIV
      IF (TMS.LE.QDIV(JDIV)) TSUB(JDIV)=FI
601   CONTINUE
      TSUB(NUMDIV)=TBURN
600   CONTINUE
      GOTO 620
610   NPWP=0
620   CONTINUE
```

```
      IF (ITYPE.GE.2) DTIME=1.
      NDIV=NUMDIV-1
      DO 700 JPWP=1,NDIV
      IF (JPWP.GT.1.AND.NPWP.EQ.0) GOTO 700
      TSTAGE=TSUB(JPWP)
      TBRN=TSUB(JPWP+1)-TSTAGE
      PCNT=QDIV(JPWP+1)-QDIV(JPWP)
      IF (NPWP.EQ.0) PCNT=1.
      IF (TBRN.LE.0.) GOTO 700
      CALL SCONST(ICAT,UW,ITYPE,FW,EFF,TBRN,NPWP,AIRT,TGRAD,C1,C2,C3,
     *XN,XLIM,YLIM,ZLIM,HLIM,TLIM,XS,CNEUT,PCNT,W)
      PCT=PCNT*100
      IF (NPWP.NE.0.AND.NOPRNT.NE.1) WRITE (IOOUT,705) PCT,TSUB(JPWP),
     *TSUB(JPWP+1)
      IF (NPWP.NE.0.AND.MCUOPT.NE.1) WRITE (IOOUT,705) PCT,TSUB(JPWP),
     *TSUB(JPWP+1)
705   FORMAT(1X,30H *** CLOUD PORTION CONTAINING ,F7.3,35H PERCENT OF SM
     *OKE DURING BURN FROM ,F7.2,3H TO,F7.2,4H SEC//)
      CL=0.
      ICODE=0
      TO=0.
      NCY=0
      ICALL=0
      ISTT=ISRN-IFIX(TM+0.0001)
      II=IFIX(TM)
      XPP0=XPP1
      I1=ISTO
      I2=IETO
      I3=IDTO
      NNN=1
      IF(MODE.EQ.0) GO TO 2000
      I1=ISTT
      I2=ISTT
      I3=1
      II=0
      NNN=NPTS
      TO=0.
2000  CONTINUE
      DO 6 J=1,NNN
      IF (ITYPE.EQ.1) TO=FLOAT(I1)-DTIME
      DO 6 I=I1,I2,I3
      L=I-II
      IF(MODE.EQ.1) XPP1=XPP0+(J-1)*DELX
      IF(MODE.EQ.1) XPP2=XPP1
      X=0.0
      Y=0.0
      Z=0.0
      PATHL=0.0
      CL=0.0
      IF(L.LT.I1) GO TO 2001
C***  TAB IS NEXT TABLE REPORT TIME.  TO IS NEXT COMPUTATION TIME.
      TAB=FLOAT(L)-TSTAGE
      IF (TAB.GT.0.) GOTO 3
      IF (NCY.GE.MAXS) GOTO 6
      NCY=NCY+1
      GOTO 6
    3 TO=TO+DTIME
      CALL WGGEOM(ICALL,CLGAUS,ITYPE,XPP1,YPP1,ZPP1,XPP2,YPP2,ZPP2,C1,
     1C2,C3,TO,UW,ICAT,HLIM,TLIM,CNEUT,XS,PATHL,X,Y,Z,XLIM,YLIM,ZLIM)
      CALL CLSMOK(ICODE,CLGAUS,ITYPE,CL,W,PATHL,TBURN,TBRN,PCNT,TSTAGE,
     *NPWP,XLIM,YLIM,ZLIM,TO,TLIM,DTIME,X,Y,Z,BRAT1,BRAT2,BRAT3,BRAT4,
     *BRAT5)
C*** REPORT OUTPUT DATA IF TIME TO .EQ. TAB, OTHERWISE, LOOP
C***  BACK FOR NEXT TIME STEP...
      IF (TO.LT.(TAB-.001)) GOTO 3
 2001 CALL STRANS(CL,SMTRAN,ITYPE,EXTC,1)
      YFULL=2.*Y
      TV=FLOAT(I)-TM
      XWRIT=FLOAT(I)
      IF(MODE.GT.0) XWRIT=XPP2+XM
```

```
      PRNTOP=NOPRNT
      IF (MODE.GT.0) PRNTOP=MCUOPT
      IF (PRNTOP.NE.1) WRITE(IOOUT,290) XWRIT,X,YFULL,Z,PATHL,CL,        SMK06070
     *(SMTRAN(K),K=1,7)                                                  SMK06080
 290  FORMAT(6X,F5.0,2X,F6.0,3X,F6.0,3X,F6.0,4X,F6.2,4X,F7.2,7(3X,F5.3))SMK06090
      IF (ICRTR.EQ.2.AND.TV.GT.(TIMTRN+.5)) GOTO 7                       SMK06100
      IF (ICRTR.EQ.1.AND.SMTRAN(KWAVE).GE.TRANS) GOTO 7                  SMK06110
      TRTM=TV                                                            SMK06120
      TRANS=SMTRAN(KWAVE)                                                SMK06130
7     CONTINUE                                                           SMK06140
      IF (NCY.GE.MAXS) GOTO 6                                            SMK06150
      NCY=NCY+1                                                          SMK06160
      CLTOT(NCY)=CLTOT(NCY)+CL                                           SMK06170
    6 CONTINUE                                                           SMK06180
700   CONTINUE                                                           SMK06190
C*** FINAL OUTPUT.                                                       SMK06200
      IF (NOMORE.EQ.0) GOTO 5                                            SMK06210
980   IF (NCY.LE.0) GOTO 999                                             SMK06220
      IF (NRUNS.LE.1.AND.NPWP.EQ.0) GOTO 997                             SMK06230
      IF(MODE.EQ.0) WRITE(IOOUT,982) NRUNS                               SMK06240
      IF(MODE.GT.0) WRITE(IOOUT,984) NRUNS
982   FORMAT(1H1,40X,19HCOMBINED EFFECT OF ,I3,21H EXECUTIONS IN SMOKE:/SMK06250
     *10X,10HTIME (SEC),5X,11HCL (G/M**2),21X,12HTRANSMISSION/           SMK06260
     *10X,10(1H-),5X,11(1H-),4X,                                         SMK06270
     *54H0.4-0.7 0.7-1.2   1.06  3.0-5.0 8.0-12.    10.6  94.GHZ/)       SMK06280
984   FORMAT(1H1,40X,19HCOMBINED EFFECT OF ,I3,21H EXECUTIONS IN SMOKE:/
     *10X,11HLOS(METERS),4X,11HCL (G/M**2),21X,12HTRANSMISSION/
     *10X,10(1H-),5X,11(1H-),4X,
     *54H0.4-0.7 0.7-1.2   1.06  3.0-5.0 8.0-12.    10.6  94.GHZ/)
      TRANS=1.0                                                          SMK06290
      DO 985 I=1,NCY                                                     SMK06300
      CL=CLTOT(I)                                                        SMK06310
      TO=FLOAT(ISTO+(I-1)*IDTO)                                          SMK06320
      CALL STRANS(CL,SMTRAN,ITYPE,EXTC,1)                                SMK06330
      YWRIT=TO
      IF(MODE.GT.0) YWRIT=XPP0+XM+(I-1)*DELX
      WRITE (IOOUT,983) YWRIT,CL,(SMTRAN(J),J=1,7)                       SMK06340
      IF(IPLT.EQ.1) WRITE(NPLOTU,883) YWRIT,CL
 883  FORMAT(1X,F6.1,2X,F8.3)
983   FORMAT(12X,F6.0,8X,F8.2,6X,7(1X,F5.3,2X))                          SMK06350
      IF (ICRTR.EQ.2.AND.TO.GT.(TIMTRN+.5)) GOTO 985                     SMK06360
      IF (ICRTR.EQ.1.AND.SMTRAN(KWAVE).GE.TRANS) GOTO 985                SMK06370
      TRTM=TO                                                            SMK06380
      TRANS=SMTRAN(KWAVE)                                                SMK06390
985   CONTINUE                                                           SMK06400
997   CONTINUE                                                           SMK06410
      WRITE (IOOUT,3100)WAVE1,TRANS,TRTM                                 SMK06420
      TRANS=SMTRAN(KWAVE)                                                SMK06430
      RETURN                                                             SMK06440
  998 WRITE (IOOUT,3200)                                                 SMK06450
      GOTO 999                                                           SMK06460
 3100 FORMAT(1H0,5X,37H***TRANSMISSION RETURNED TO MAIN FOR              SMK06470
     1           14HWAVELENGTH OF ,F8.3,16H MICROMETERS IS ,F5.3,        SMK06480
     2           8H AT TIME ,F7.0)                                       SMK06490
 3200 FORMAT(1H0,10X,35HINVALID WAVELENGTH PASSED FROM MAIN              SMK06500
     1           ,//,10X,27H TRANS=1.0 RETURNED TO MAIN,//)              SMK06510
 310  CONTINUE                                                           SMK06520
C***  ERROR CONDITION                                                    SMK06530
      WRITE(IOOUT,320)                                                   SMK06540
 320  FORMAT(1H ,105HMORE THAN 13 DATA CARDS HAVE BEEN INPUT . PLEASE CHSMK06550
     *ECK THAT THERE ARE NO MORE THAN 13 DATA CARDS PER RUN.)            SMK06560
  999 CONTINUE                                                           SMK06570
      IERR=1                                                             SMK06580
      TRANS=1.0                                                          SMK06590
       RETURN                                                            SMK06600
      END                                                                SMK06610
```

```
      SUBROUTINE CLSMOK(ICODE,CLGAUS,ITYPE,CL,W,PATHL,TBURN,TBRN,PCNT,
     *TSTAGE,NPWP,XLIM,YLIM,ZLIM,TO,TLIM,DTIME,X,Y,Z,BRAT1,BRAT2,
     *BRAT3,BRAT4,BRAT5)
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C
C NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION
C       AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)
C
      COMMON /MO5/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9),
     *EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)
C
C  NOTE: TO CHANGE MAXIMUM BURN OR OBSCURATION DURATION, CHANGE
C      PARAMETER MAXS AND DIMENSIONS OF SMAS,PVOL,CLTOT TO MAXS
C
C     THIS SUBROUTINE CALCULATES THE VOLUME OF THE SMOKE CLOUD AND
C     THE CONCENTRATION * LENGTH(CL) AT THE INTERSECTION OF THE OBSERVER
C     TARGET LOS OF THE CLOUD FORMED BY THE TOTAL NUMBER OF SMOKE
C     MUNITIONS WHICH WERE DETONATED AT THE SAME TIME AND FROM THE SAME
C     POINT.  FOR CONTINUOUS TYPE BURNS (HC,FOG OIL, PWP, RP, AND
C     WP WICKS OR WEDGES), THE BURN MASS INCREMENT AND PATH LENGTH
C     TO VOLUME RATIO ARE STORED FOR EACH PUFF.  THESE PUFFS ARE
C     SUBSEQUENTLY ADDED TO FIND THE TOTAL EFFECT OF OBSCURANT.
C
C        INPUTS
C
C               ICODE   A FLAG TO BE SET BY USER TO 0 ON FIRST CALL
C                       TO PROGRAM, WHICH WILL THEN RESET IT TO 1.
C               ITYPE   SMOKE TYPE   1=WP,   2=PWP OR WP WICK/WEDGE
C                                    3=HC,  4=FOG OIL,  5=RP
C               W       SMOKE MASS PRODUCED BY XN  MUNITIONS (G FOR
C                       TYPES 1-3 AND 5, G/S FOR TYPE 4)
C               PATHL    PATHLENGTH OF SMOKE CLOUD AS IT INTERSECTS THE
C                        OBSERVER-TARGET LINE OF SIGHT  (M)
C               TBURN    TOTAL LENGTH OF TIME OF BURN (S)
C               NPWP     FLAG FOR PWP/RP/WP WICKS OR WEDGES. IF NON-ZERO,
C                        THEN BURN IS IN 25 PERCENT (OF W) STAGES.
C               TBRN     PARTIAL BURN DURATION THIS STAGE (SEC)
C               TSTAGE   START OF PARTIAL BURN (SEC)
C               XLIM,YLIM,ZLIM DIMENSIONS (LENGTH,HALF-WIDTH,HEIGHT)
C                        OF CLOUD AT END OF EXOTHERMIC RISE TIME (M)
C               TO      TIME AFTER IGNITION (S)
C               TLIM    TIME OF TERMINATION OF HEAT RISE (S)
C               DTIME   TIME INCREMENT OF COMPUTATION (S)
C               ENDTIM  ENDING TIME OF COMPUTATION (S)
C               X,Y,Z   CURRENT LEADING EDGE CO-ORDINATES (M)
C               BRAT1,2,3,4 POLYNOMIAL BURN RATE COEFF.
C               CLGAUS   UNIT CONTRIBUTION FROM GAUSSIAN PUFF.
C        OUTPUTS
C               CL      COMPUTED CL IN (G/M**2) FOR THIS MUNITION SET.
C
C*** SIMPLE FOR INSTANTANEOUS BURN OF WP
      BFUN(T)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*BRAT4/4.)))+BRAT5*
     *0.07818288*ALOG(1.+358800.*T)
      MAXS=960
      IF (ITYPE.GT.1) GOTO 100
      VOL=0.25*(4.*PI/3.) *X*Y*Z
      CL=W*(.25*PATHL/VOL  +.75  *CLGAUS)
      RETURN
C*** INITIALIZE
  100 CL=0.
      IF (ICODE.NE.0) GOTO 110
      ICODE=1
      IB=0
      IPL=0
      START=0.
      BRNOUT=FLOAT(IFIX(.9999+TBRN))
      NRAT=2
      IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0.) NRAT=1
      IF (BRAT5.NE.0.) NRAT=5
```

```
      KSUB=0
      TOTPV=0.
      BR=W
      IF (ITYPE.NE.4) BR=W/BRNOUT
      IF (NPWP.NE.0) BR=BR*PCNT
      IF (BRNOUT.LE.1.) NRAT=1
      IF (NRAT.GT.1) BTHEN=BFUN(TSTAGE/TBURN)
C*** STORE PUFF MASS EMITTED AT TIME TO
110   IF (TO.GT.BRNOUT) GOTO 120
      IF (IB.GE.MAXS) GOTO 120
      IB=IB+1
      IF (NRAT.EQ.1) GOTO 120
      IF (IB.EQ.MAXS) WRITE (IOOUT,900) IB
900   FORMAT(1X,61H*** WARNING - IN SMOKE, MAXIMUM STORAGE FOR BURN DURA
     *TION OF ,I5,55H SEC IS FULL. ACCURACY BEYOND THIS POINT DECREASES.
     * ***  )
      TM=TO
      IF (TO.GT.TBRN) TM=TBRN
      T=(TM+TSTAGE)/TBURN
      BNOW=BFUN(T)
      SMAS(IB)=W*(BNOW-BTHEN)
      BTHEN=BNOW
      IF (ITYPE.EQ.4) SMAS(IB)=SMAS(IB)*TBURN
C*** COMPUTE VOLUME AT TIME TO OF FIRST PUFF
120   IF (ITYPE.EQ.3) GOTO 130
      IF (ITYPE.EQ.4) GOTO 140
C*** PWP, RP OR WP WICKS/WEDGES CLOUD
      VOL=0.25*(4.*PI/3.)*X*Y*Z
      GOTO 200
130   IF (TO.LE.TLIM) GOTO 140
C*** POST RISE REGION HC CONE.
      VOL=0.5*(PI/3.)*XLIM*YLIM*ZLIM
C*** POST-RISE FRUSTRUM OF APPROXIMATED ELLIPTIC CONE.
      XZPROJ=Z*(X-XLIM)/(Z-ZLIM)
      VFRUST=0.5*(PI/3.)*(XZPROJ*(Y*Z-YLIM*ZLIM)+(X-XLIM)*YLIM*ZLIM)
      VOL=VOL+VFRUST
      GOTO 200
C*** HC BEFORE THE END OF EXOTHERMIC RISE AND FOG OIL CASE
140   VOL=0.5*(PI/3.)*X*Y*Z
C*** STORE PATH LENGTH/VOLUME RATIO AT TIME TO FOR FIRST EXPANDING
C    UNIFORM AND GAUSSIAN PUFF CONTRIBUTION.
200   IF (ITYPE.NE.4) PV=.25*(PATHL/VOL)+.75*CLGAUS
      IF (ITYPE.EQ.4) PV=PATHL/VOL
      IF (PV.LE.0..AND.START.EQ.0.) RETURN
      IF (PV.GT.0..AND.START.EQ.0.) START=TO
      IF (IPL.GE.MAXS) GOTO 300
      IPL=IPL+1
      IF (IPL.EQ.MAXS) WRITE (IOOUT,910) TO,IPL
910   FORMAT(1X,58H*** WARNING - IN SMOKE, MAX. STORAGE FOR CLOUD VOLUME
     *S OF ,I5,34H SEC. IS FULL. COMPUTATION TIME = ,F6.0,5H SEC./
     *15X,40HACCURACY BEYOND THIS POINT DECREASES ***)
      PVOL(IPL)=PV
C*** SUM CL FOR PUFFS
300   IF (NRAT.EQ.1) GOTO 400
      LMIN=MAX0(1,(IPL-IB+1))
      IF (IPL.LT.LMIN) RETURN
      N=0
      DO 320 J=LMIN,IPL
      K=IPL-J+1
      CDEL=SMAS(K)*PVOL(J)
      CL=CL+CDEL
      N=N+1
      IF (N.LT.120) GOTO 320
      IF (ABS(CDEL).LT.1.E-5) RETURN
320   CONTINUE
      RETURN
C*** FAST COMPUTATION FOR CONSTANT BURN RATES
400   TOTPV=TOTPV+PVOL(IPL)
C INDEX OF LAST PUFF EMITTED
      IF (IPL.LT.MAXS) KSUB=IPL-IB
```

```
C KEEP REMOVING LAST PUFF IF POSSIBLE
      IF (IPL.EQ.MAXS) KSUB=KSUB+1
      IF (KSUB.GT.IPL) KSUB=IPL
C ONLY REMOVE PUFFS AFTER BURN HAS STOPPED (IE LAST PUFF OUT)
      IF (KSUB.GT.0) TOTPV=TOTPV-PVOL(KSUB)
      CL=TOTPV*BR
      RETURN
      END
```

```
      SUBROUTINE SCONST(ICAT,UW,ITYPE,FW,EFF,TBRN,NPWP,AIRT,TGRAD,C1,C2,
     *       C3, XN,XLIM,YLIM,ZLIM,HLIM,TLIM,XS,CNEUT,PCNT,W)
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C
C     SMOKE MODEL  PARAMETERS FOR CLOUD DIMENSIONS
C     THIS SUBROUTINE CALCULATES THE PARAMETERS FOR THE CLOUD WIDTH AND
C     CLOUD HEIGHT(MOMENTUM) AS A FUNCTION OF PASQUILL CATEGORY AND THE
C     HEAT RISE PARAMETER FOR THE HEIGHT(EXOTHERMIC). HEAT RISE IS
C     TERMINATED AS A FUNCTION OF ATMOSHERIC STABILITY
C     INPUTS    ICAT    PASQUILL CATEGORY
C                           1-A, 2-B, 3-C, 4-D, 5-E, 6-F
C               UW      WIND VELOCITY  (M/S)
C               ITYPE   TYPE OF SMOKE.  1=WP, 2=PWP OR WP WICK/WEDGE,
C                                       3=HC, 4=FOG OIL, 5=RP
C               FW      FILL WEIGHT  (LBS)
C               EFF     EFFICIENCY (PERCENT)
C               XN      NUMBER OF MUNITIONS
C               TBRN    BURN TIME FOR THIS STAGE (OR ENTIRE MUNITION)SEC.
C               NPWP    FLAG FOR LONG-BURN PHOSPHORUS. IF NON-ZERO, THEN
C                       BURNS ARE IN 25 PERCENT STAGES TO ALLOW SEPARATE
C                       BUOYANCIES FOR EACH.
C               AIRT    SURFACE AIR TEMPERATURE     (DEG C)
C               TGRAD   VERT TEMP GRADIENT (C DEG./M) USED ONLY FOR
C                            CATAGORIES E,F (IE. 5,6) IN WHICH IT MUST
C                            BE POSITIVE.
C     OUTPUTS   C1      EQUATION PARAMETER FOR CLOUD WIDTH  Y=9.1+C1*X
C               C2      PARAMETER FOR HEIGHT (MOM)  Z=2.73+C2*X
C               C3      PARAMETER FOR HEIGHT (EXO)  Z=2.73+C2*X+C3*X**2/3
C               XLIM    CLOUD LENGTH ALONG WIND DIRECTION (M)
C                         AT RISE TERMINATION.
C               YLIM    CLOUD BASE HALF-WIDTH PERP. WIND DIRECTION (M)
C                         AT RISE TERMINATION.
C               ZLIM    TOTAL CLOUD HEIGHT AT TERMINATION OF HEAT RISE.
C               HLIM    TOTAL ADDED RISE AT TERMINATION OF HEAT RISE.
C               TLIM    TIME OF TERMINATION OF HEAT RISE
C               XS      TERM FOR CALCULATION OF HEIGHT FOR NEUTRAL CONDITI
C               CNEUT   TERM FOR CALCULATION OF HEIGHT FOR NEUTRAL CONDITI
C
      DIMENSION C11(6),C22(6)
      DATA C11/.419,.328,.238,.2,.18,.146/
      DATA C22/.137,.11,.073,.066,.055,.046/
      TLIM=600.
      XS=0.0
      CNEUT=0.0
      XLIM=0.
      YLIM=0.
      ZLIM=0.
      HLIM=1.0
C***  ERROR CONDITION
      IF(ICAT.LT.1.OR.ICAT.GT.6)   WRITE(IOOUT,50)
   50 FORMAT(1H ,62HERROR IN SUBROUTINE SCONST. PASQUILL CATEGORY IS NOT
     1ACCEPTABLE)
      IF (ICAT.LT.1.OR.ICAT.GT.6) RETURN
C***  SELECTION OF CLOUD WIDTH COEFFICIENT AS A FUNCTION OF PASQUILL
C***  CATEGORY.
      C1 = C11(ICAT)
C***  SELECTION OF CLOUD HEIGHT(MOM) COEFFICIENT AS A FUNCTION OF
C***  PASQUILL CATEGORY.
      C2 = C22(ICAT)
      IF(ICAT.GT.1)GO TO 1
      IF(UW.GT.2.0)GO TO 1
      C2=0.15
      IF(UW.GT.1.5)GO TO 1
      C2=0.18
      IF(UW.GT.1.0)GO TO 1
      C2=0.25
      IF(UW.GT.0.5)GO TO 1
      C2=0.39
    1 CONTINUE
```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

2 6

```
 120  TB=TBRN
      IF (ITYPE.EQ.1.AND.TBRN.GT.1.) TB=1.
      IF (NPWP.NE.0) TB=TB/PCNT
C***  IF NON-EXOTHERMIC, RETURN.
      IF (ITYPE.EQ.4) RETURN
      EM=900.0
      IF(ITYPE.EQ.3)EM=500.0
      IF(ITYPE.EQ.5)EM=660.
      F=0.00001*3.59*453.59*EM*FW*XN*(EFF/100.)/TB
C***  CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT FOR UNSTABLE ATM.
C***  CONDITIONS (A,B,C)
      C3=1.6/UW*(F**.3333)
      IF(ITYPE.EQ.3) GO TO 200
      IF(ICAT.EQ.4) GO TO 131
      IF(ICAT.GT.4) GO TO 141
      TLIM=0.0
C***  DETERMINE TIME AND TOTAL CLOUD HEIGHT AT TERMINATION OF HEAT RISE.
 123  TLIM=TLIM+2.
      X=UW*TLIM
      Y=(9.1+C1*X)/2.
      V=.75*(2.*3.14159)**1.5*(Y/2.15)**2*(2.73+C3*X)/4.3
      C=W/(V*TB)
      IF(C.GT.0.11) GO TO 123
      HLIM=C3*X**.667
      RETURN
C***  CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENTS FOR NEUTRAL ATM.
C***  CONDITIONS (D).
 131  XS=10.0*F**0.4
      CNEUT=1.6*(F**.3333)*XS**0.667/UW
      RETURN
 141  CONTINUE
C***  CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT AND TOTAL CLOUD
C***  HEIGHT AT TERMINATION OF HEAT RISE FOR STABLE ATM. CONDITIONS(E,F)
      SBAR=9.8/(AIRT+273.0)*(TGRAD+0.0098)
      HLIM=1.4*(F/(UW*SBAR))**.333
      RETURN
C *** CALCULATE PARAMETERS FOR HC 10 SEC. RISE TIME.
  200 TLIM=10.
      XLIM=TLIM*UW
      YLIM=(9.1+C1*XLIM)/2.
      ZLIM=2.73+C2*XLIM
      IF (ICAT.EQ.4) GOTO 231
      HLIM=C3*XLIM**.667
      ZLIM=ZLIM+HLIM
      IF (ICAT.LT.5) RETURN
      SBAR=9.8/(AIRT+273.)*(TGRAD+0.0098)
      HTEST=1.4*(F/(UW*SBAR))**.333
      IF (HLIM.LE.HTEST) RETURN
C***  IF UNSTABLE ATMOSPHERE REACHES MAX. BEFORE 10 SEC. COMPUTE
C***  TLIM.
      XLIM=(HTEST/C3)**1.5
      TLIM=XLIM/UW
      YLIM=(9.1+C1*XLIM)/2.
      HLIM=HTEST
      ZLIM=2.73+C2*XLIM+HLIM
      RETURN
  231 XS=10.0*F**0.4
      CNEUT=C3*(XS**.667)
      HLIM=CNEUT*(0.4+0.64*(XLIM/XS)+2.2*(XLIM/XS)**2)/
     *(1.+0.8*XLIM/XS)**2
      ZLIM=ZLIM+HLIM
      RETURN
      END
```

```
      SUBROUTINE SMASSP(XN,EFF,FW,RELHUM,W,ITYPE,YF)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C
C     SMOKE MODEL
C     THIS SUBROUTINE CALCULATES THE SMOKE MASS PRODUCED BY XN
C     MUNITIONS AT THE SAME POINT AND SAME TIME.
C     INPUTS  ITYPE SMOKE TYPE
C                         1 WHITE PHOSPHOROUS
C                         2 PLASTICIZED WHITE PHOSPHORUS OR WP WICK WEDGE
C                         3 HC
C                         4 FOG OIL
C                         5 RED PHOSPHORUS
C             XN     NUMBER OF MUNITIONS AT THIS POINT AND TIME.
C             RELHUM RELATIVE HUMIDITY (PERCENT)
C             EFF    MUNITION EFFICIENCY (PERCENT)
C             FW     FILL WEIGHT  (LBS FOR TYPES 1-3, GAL/HR FOR TYPE 4)
C             W      SMOKE MASS PRODUCED FOR XN MUNITIONS.
C                         G FOR TYPES 1-3 AND 5, G/S FOR TYPE 4
C             YF     OPTIONAL USER SUPPLIED YIELD FACTOR
C
      W=0.0
      IF(ITYPE.LT.1.OR.ITYPE.GT.5) GO TO 90
      GO TO (101,101,102,103,101),ITYPE
   90 WRITE(IOOUT,50)ITYPE
   50 FORMAT(//,28H ERROR IN SMASSP:SMOKE TYPE ,I4,10H UNDEFINED)
      RETURN
  101 Y=.028*RELHUM+3.4
      CONVER=453.592
      GO TO 110
  102 Y=1.17 +.014*RELHUM
      CONVER=453.592
      GO TO 110
C***  FOR FOG OIL, W IS A RATE, WHERE FW IS IN GAL/HR AND .93
C***  CONVERTS TO G/S.
  103 Y=1.0
      CONVER=0.93
110   IF (YF.NE.0.) Y=YF
      YF=Y
      W=XN*Y*FW*CONVER*(EFF/100.0)
      RETURN
      END
```

```
      SUBROUTINE STRANS(CL,SMTRAN,ITYPE,EXTC,ICALL)
C
C
C     SMOKE MODEL     CLOUD TRANSMISSION
C     THIS SUBROUTINE CALCULATES TRANSMISSION IN 7 SPECTRAL REGIONS
C   (0.4-0.7,0.7-1.2,1.06,3.0-5.0,8.0-12.0,10.6 MICROMETERS AND 94.GHZ)
C     FOR A GIVEN SMOKE TYPE AND CONCENTRATION LENGTH
C     INPUTS    ITYPE   SMOKE TYPE
C                          1   WHITE PHOSPHOROUS
C                          2   PLASTICIZED WHITE PHOSPHORUS, WP WICK WEDGE
C                          3   HC
C                          4   FOG OIL
C                          5   RED PHOSPHORUS
C               CL      COMPUTED CL IN (G/M**2)
C               ICALL      = 0 SETS UP EXTC ARRAY USED FOR COMPUTATIONS AND
C                            ALLOWS USER TO OVER-RIDE DEFAULT ALPHAS.
C                          = 1 EXECUTES TRANSMISSION CALCULATION.
C
C     OUTPUTS   SMTRAN TRANSMISSION THROUGH SMOKE (DECIMAL)
C               EXTC  ARRAY OF EXTINCTION COEFF. ACTUALLY USED
C                     IN TRANSMISSION CALCULATION. EXTC(8) IS
C                     USED AS A FLAG FOR ICALL=0 REPLACEMENT.
C
C               IF EXTC(8)=ITYPE, NO CHANGES ARE MADE IN EXTC ARRAY IE.,
C                          ALPHA VALUES DO NOT REPLACE EXTC VALUES.
C               IF EXTC(8)=0.      , ONLY THOSE VALUES IN EXTC WHICH ARE
C                          ZERO ARE REPLACED BY THE STORED VALUES IN
C                          ALPHA (ITYPE COLUMN).
C               IF EXTC(8) IS NOT 0. OR ITYPE, THEN ALL EXTC VALUES
C                          ARE REPLACED BY CORRESPONDING ALPHA VALUES AND
C                          EXTC(8) IS SET TO ITYPE.
C
      DIMENSION ALPHA(7,5),SMTRAN(7),EXTC(8)
      DATA ALPHA           /4.304,2.166,1.541,0.350,0.338,0.364,0.001,
     *                      4.304,2.166,1.541,0.350,0.338,0.364,0.001,
     *                      4.579,2.186,2.040,0.190,0.052,0.051,0.001,
     *                      6.851,4.592,3.497,0.245,0.020,0.018,0.001,
     *                      4.304,2.166,1.541,0.350,0.338,0.364,0.001/
C***  TRANSMISSION CALCULATED BY BEER'S LAW APPROXIMATION
C***    IF ICALL=0, EXTC ARRAY IS FORMED OR MODIFIED...
      IF (ICALL.NE.0) GOTO 20
      IF (EXTC(8).EQ.FLOAT(ITYPE)) GOTO 18
      IF (EXTC(8).EQ.0.) GOTO 15
      DO 13 J=1,7
   13 EXTC(J)=0.
   15 EXTC(8)=FLOAT(ITYPE)
      DO 17 J=1,7
      IF (EXTC(J).EQ.0.) EXTC(J)=ALPHA(J,ITYPE)
   17 CONTINUE
   18 RETURN
C***  FOR ICALL NON-ZERO, COMPUTE TRANSMISSION USING EXTC VALUES.
   20 DO 30 I=1,7
      SMTRAN(I)=EXP(-EXTC(I)*CL)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WGGEOM(ICALL,CLGAUS,ITYPE,XPP1,YPP1,ZPP1,XPP2,YPP2,
     1ZPP2,C1,C2,C3,TO,UW,ICAT,HLIM,TLIM,CNEUT,XS,PATHL,X,Y,Z,XLIM,
     2YLIM,ZLIM)
C
C NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION
C       AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)
C
      COMMON /M05/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9),
     *EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA MAXS /960/
C
C  NOTE: TO CHANGE LENGTH OF BURN OR OBSCURATION, RESET
C       MAXS AND REDIMENSION SMAS, PVOL AND CLTOT ARRAYS
C
C   SMOKE MODEL   GEOMETRY
C     INPUT:
C     ITYPE = SMOKE TYPE CODE
C     XPP1,YPP1,ZPP1 = OBSERVER IN MUNITION CENTERED COORDINATES
C     YPP1,YPP2,ZPP2 = TARGET IN MUNITION CENTERED COORD. (X AXIS ALONG
C                          WIND VECTOR.)
C     C1,C2,C3 = CLOUD GROWTH PARAMETERS (SEE PROGRAM SCONST)
C     TO = TIME SINCE DETONATION (SECONDS)
C     UW = WIND SPEED (METERS/SECOND)
C     ICAT=PASQUILL CATEGORY
C     HLIM,TLIM,CNEUT,XS= EXOTHERMIC RISE PARAMETERS
C     ICALL = FLAG SET BY USER TO 0 FOR 1ST CALL. RESET BY PROGRAM
C             TO 1 THEREAFTER.
C    OUTPUT:
C    X = CLOUD LENGTH (METERS)
C    Y = CLOUD HALF-WIDTH (METERS)
C    Z = CLOUD HEIGHT (METERS)
C    PATHL = PATHLENGTH OF LOS THROUGH SMOKE CLOUD
C
C  SUBROUTINES CALLED... DIRECTLY: XYZINT, GPUFF, INDIRECTLY: QROOT
C
C *** TRANSLATION TO PLACE MUNITION AT (0,0,0)
      IF (ICALL.NE.0) GOTO 10
      ICALL=1
      KCALL=0
C***  CALCULATE LEADING EDGE LOCATION AT TO.
10    A=UW*TO
      B=(9.1/2.0)+A*C1/2.
      IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) GOTO 33
C***  WP/PWP/RP COMPUTATION.
      GO TO (11,11,11,21,31,31),ICAT
 11   CONTINUE
      C=2.73+C2*A+C3*A**.667
      CLIM=2.73+C2*A+HLIM
      IF(TO.GT.TLIM.AND.C.GT.CLIM)C=CLIM
      GO TO 41
 21   CONTINUE
      C=2.73+C2*A
      IF (A.LE.XS) C=C+C3*A**.667
      IF (A.GT.XS) C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/
     1(1.0+0.8*(A/XS))**2
      GO TO 41
 31   CONTINUE
      C=2.73+C2*A+C3*A**0.667
      CLIM=2.73+C2*A+HLIM
      IF(C.GT.CLIM)C=CLIM
      GOTO 41
C*** HC, FOG OIL COMPUTATION.
   33 C=2.73+C2*A
      IF (ITYPE.EQ.4) GOTO 41
      IF (TO.LT.TLIM) GOTO 35
      C=C+HLIM
      GOTO 41
   35 IF (ICAT.NE.4.OR.A.LE.XS)C=C+C3*A**0.667
      IF (ICAT.EQ.4 .AND. A.GT.XS)
```

```
     1 C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/
     1(1.0+0.8*(A/XS))**2
   41 CONTINUE
C*** NOW COMPUTE PATH LENGTHS -- SELECT ELLIPSOID FOR PHOSPHORUS
C*** OR ELLIPTICAL CONES FOR HC, FOG OIL.
C***  FIRST GAUSSIAN PUFF AND IMAGE PUFF CONTRIBUTION, THEN FULLY MIXED
      NCODE=2
      IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) NCODE=1
      IF (ITYPE.NE.4)
     *CALL GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1,ZPP2)
      X=A
      Y=B
      Z=C
      IF (ITYPE.EQ.3) GOTO 200
100   X0=A
      Y0=B
      Z0=C
150   XL(1)=XPP1
      XL(2)=XPP2
      YL(1)=YPP1
      YL(2)=YPP2
      ZL(1)=ZPP1
      ZL(2)=ZPP2
      CALL XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)
      PATHL=SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+
     *(ZINT(1)-ZINT(2))**2)
      IF (ITYPE.EQ.3) GOTO 220
      RETURN
200   ICONE=1
      IF (TO.LE.TLIM) GOTO 100
      ICONE=0
C*** FOR HC, BREAK PATH UP INTO TWO PARTS. FIRST RISE PORTION CONE,
C    THEN POST-RISE FRUSTRUM.
      X0=XLIM
      Y0=YLIM
      Z0=ZLIM
      GOTO 150
220   IF (ICONE.EQ.1) RETURN
      ICONE=1
C*** NOW POST RISE PORTION,
      IF (PATHL.EQ.0.) GOTO 100
      IF (IFLAG(1).NE.3.AND.IFLAG(2).NE.3) RETURN
      IF (IFLAG(1).EQ.3.AND.IFLAG(2).EQ.3) RETURN
      K=1
      IF (IFLAG(2).EQ.3) K=2
      J=1
      IF (XL(2).LT.XL(1)) J=2
      XL(J)=XINT(K)
      YL(J)=YINT(K)
      ZL(J)=ZINT(K)
      X0=A
      Y0=B
      Z0=C
      CALL XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)
      PATHL=PATHL+SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+
     *(ZINT(1)-ZINT(2))**2)
      RETURN
      END
```

```
      FUNCTION QROOT(ISIGN,A,B,C)
C*****PURPOSE:
C            TO FIND ROOTS OF A QUADRATIC EQUATION.
      IF (A.EQ.0.) GOTO 2
      XX=1.0*ISIGN
      TEST= B*B - 4.0*A*C
      IF(TEST.LT.0.0)GO TO 1
      QROOT=(-1.0*B + XX*SQRT(TEST))/(2.0*A)
      GO TO 100
    1 QROOT=0.0
      GOTO 100
    2 IF (B.EQ.0.) GOTO 1
      QROOT=-C/B
  100 RETURN
      END
```

```
      SUBROUTINE XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)
      DIMENSION XL(2),YL(2),ZL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)
      DIMENSION DIST(2),TEST(2)
C*****SUBROUTINE XYZINT*****
C*****PURPOSE:
C          TO FIND THE X,Y,Z INTERCEPTS OF A TARGET-OBSERVER LINE OF SIGHT
C          WITH A SMOKE CLOUD DESCRIBED BY:
C              NCODE = 1 A HALF ELLIPTIC CONE WITH APEX AT THE ORIGIN
C                        AND LEADING EDGE TRUNCATION BY THE X=X0 PLANE
C                        AND BOTTOM EDGE TRUNCATION BY THE Z=0 PLANE.
C              NCODE = 2 A QUARTER ELLIPSOID  WITH APEX
C                     AT THE ORIGIN AND WITH LEADING EDGE TRUNCATED BY
C                     THE X=X0 AND BOTTOM EDGE TRUNCATED BY THE Z=0 PLANE.
C*****INPUT:
C          XL(2),YL(2),ZL(2) = X,Y,Z COORDINATES OF TWO POINTS THROUGH
C          WHICH THE LOS PASSES(IE. TARGET AND OBSERVER COORDINATES).
C          X0,Y0,Z0 = LENGTH OF SEMI-AXES OF ELLIPSOID.
C*****OUTPUT:
C          XINT(2),YINT(2),ZINT(2) = X,Y,Z COORDINATES OF THE INTERCEPTS
C          OF THE LOS WITH THE ELLIPSOID.
C          IFLAG(2) - INTERCEPT TYPE FOR EACH INTERCEPT COORD:
C                   = 0  NO INTERCEPT
C                   = 1  INTERIOR TO VOLUME
C                   = 2  ON CONICAL OR ELLIPTICAL SURFACE
C                   = 3  ON LEADING EDGE OF SURFACE
C*****MATHEMATICAL APPROACH:
C          THE EQUATION OF THE ELLIPSOID CAN BE WRITTEN AS:
C          ((X-X0)/X0)**2 + (Y/Y0)**2 +(Z/Z0)**2 = 1
C          AND THE EQUATION OF THE LOS CAN BE WRITTEN AS:
C          (X-XL1)/(XL2-XL1) = (Y-YL1)/(YL2-YL1) = (Z-ZL1)/(ZL2-ZL1)
C          THE TWO EQUATIONS ARE COMBINED TO FORM A QUADRATIC EQUATION
C          WHICH IS SOLVED TO GIVE THE INTERCEPTS.
C          SIMILARLY FOR THE LOS EQUATIONS AND ELLIPTIC CONE :
C            (Z/Z0)**2 + (Y/Y0)**2 - (X/X0)**2 = 0
C*****SPECIAL NOTES
C     (1) WHEN TWO OR MORE COORDINATES ARE THE SAME, SPECIAL CASES ARE
C          FORMED WHICH MUST BE DEALT WITH SEPARATELY BECAUSE OF
C          SINGULARITIES IN THE LOS EQUATION.
C     (2) WHEN TARGET AND/OR OBSERVER ARE INSIDE THE CLOUD INTERCEPTS
C          ARE TAKEN AS THE TARGET AND/OR OBSERVER COORDINATES.
C     (3) PROPER ACCOUNT IS TAKEN FOR A LOS INTERCEPTING THE CLOUD
C          LEADING EDGE BUT.........
C     (4) ALL COORDINATES MUST BE ABOVE THE Z=0 PLANE (IE. ABOVE THE
C          SURFACE.)
C
C  SUBROUTINE CALLED... QROOT
C*****INITIALIZE PROGRAM VARIABLES
      I1=0
      I2=0
      INT=0
      LEAD=0
      ISURF=0
      K1=0
      K2=0
      K3=0
      K4=0
      TEST(1)=0.0
      TEST(2)=0.0
      TEST3=0.0
      TEST4=0.0
      DIST(1)=0.0
      DIST(2)=0.0
      DELX=XL(2)-XL(1)
      DELY=YL(2)-YL(1)
      DELZ=ZL(2)-ZL(1)
      IFLAG(1)=0
      IFLAG(2)=0
C*** REJECT IMMEDIATELY IF BOTH TGT/OBS BELOW Z=0.
      IF (ZL(1).LT.0..AND.ZL(2).LT.0.) GOTO 800
C*****DETERMINE SPECIAL CASES FOR LOS
```

```
      ICASE=1
      IF (ABS(DELX).GT.(.01*ABS(DELY)).AND.
     *    ABS(DELX).GT.ABS(.01*ABS(DELZ))) GOTO 12
      ICASE=2
      IF (ABS(DELY).GT.(.01*ABS(DELZ))) GOTO 12
      ICASE=3
      IF (DELZ.GT.1.E-2.OR.DELZ.LT.-1.E-2) GOTO 12
      ICASE=4
   12 CONTINUE
C*****DEFAULT SPECIAL CASE OF OBS-TAR COINCIDENT
      IF(ICASE.NE.4)GO TO 14
      GO TO 800
   14 CONTINUE
C*****SET UP TEST TO DETERMINE IF TARGET AND/OR OBSERVER ARE IN THE
C     INTERIOR OF THE CLOUD
      DO 1 I=1,2
      IF (NCODE.EQ.2) TEST(I)= ((XL(I)-X0)/X0)**2 + (YL(I)/Y0)**2 +
     *(ZL(I)/Z0)**2-1.
      IF (NCODE.EQ.1) TEST(I)=(YL(I)/Y0)**2 + (ZL(I)/Z0)**2 -
     *(XL(I)/X0)**2
    1 CONTINUE
      IF(TEST(1).GT.0.0)GO TO 2
      IF (XL(1).LT.0..OR.ZL(1).LT.0.) GOTO 2
      IF (XL(1).GT.X0) GOTO 2
      IF (XL(1).EQ.X0) K1=1
      IF (TEST(1).EQ.0..OR.ZL(1).EQ.0.) K3=1
      I1=1
    2 CONTINUE
      IF(TEST(2).GT.0.0)GO TO 3
      IF (XL(2).LT.0..OR.ZL(2).LT.0.) GOTO 3
      IF (XL(2).GT.X0) GOTO 3
      IF (XL(2).EQ.X0) K2=1
      IF (TEST(2).EQ.0..OR.ZL(2).EQ.0.) K4=1
      I2=1
C*****IF BOTH POINTS ARE IN THE CLOUD SET INTERCEPTS EQUAL TO THE
C     TARGET-OBSERVER COORDINATES AND RETURN--OTHERWISE CONTINUE
    3 IF(I1.EQ.0)GO TO 4
      IF(I2.EQ.0)GO TO 4
      DO 5 I=1,2
      IFLAG(I)=1
      XINT(I)=XL(I)
      YINT(I)=YL(I)
    5 ZINT(I)=ZL(I)
      IF (K3.EQ.1) IFLAG(1)=2
      IF (K4.EQ.1) IFLAG(2)=2
      IF (K1.EQ.1) IFLAG(1)=3
      IF (K2.EQ.1) IFLAG(2)=3
      GO TO 999
C*****IF ONLY ONE POINT IS IN CLOUD KEEP TRACK OF IT FOR LATER
    4 CONTINUE
      IF(I1.EQ.0)GO TO 6
      INT=1
    6 IF(I2.EQ.0)GO TO 7
      INT=2
    7 CONTINUE
      IF (K1.EQ.1) LEAD=1
      IF (K2.EQ.1) LEAD=2
      IF (K3.EQ.1) ISURF=1
      IF (K4.EQ.1) ISURF=2
C*****SET UP LOS EQUATION DEPENDING UPON CASE
      GO TO (10,20,30),ICASE
C*****CASE 1
   10 SX=DELX/DELX
      SY=DELY/DELX
      SZ=DELZ/DELX
      XI=XL(1)-SX*XL(1)
      YI=YL(1)-SY*XL(1)
      ZI=ZL(1)-SZ*XL(1)
      GO TO 101
C*****CASE 2
```

```
   20 SX=DELX/DELY
      SY=DELY/DELY
      SZ=DELZ/DELY
      XI=XL(1)-SX*YL(1)
      YI=YL(1)-SY*YL(1)
      ZI=ZL(1)-SZ*YL(1)
      GO TO 101
C*****CASE 3
   30 SX=DELX/DELZ
      SY=DELY/DELZ
      SZ=DELZ/DELZ
      XI=XL(1)-SX*ZL(1)
      YI=YL(1)-SY*ZL(1)
      ZI=ZL(1)-SZ*ZL(1)
  101 CONTINUE
C*****SET UP QUADRATIC COOEFICIENTS
      IF (NCODE.EQ.1) GOTO 60
      A=(SX/X0)**2 + (SY/Y0)**2 + (SZ/Z0)**2
      B=2.0*((XI/X0)*(SX/X0) + (YI/Y0)*(SY/Y0) + (ZI/Z0)*(SZ/Z0)
     *-(SX/X0))
      C=(XI/X0)**2 + (YI/Y0)**2 +(ZI/Z0)**2 - 2.0*XI/X0
      GOTO 61
   60 A=(SY/Y0)**2 + (SZ/Z0)**2 -(SX/X0)**2
      B=2.0*((SY/Y0)*(YI/Y0) + (SZ/Z0)*(ZI/Z0) + (SX/X0)*(XI/X0))
      C=(YI/Y0)**2 + (ZI/Z0)**2 -(XI/X0)**2
   61 CONTINUE
C*****DEFAULT ALL INTERCEPTS IF ROOTS ARE COMPLEX
      TEST0=B*B-4.0*A*C
      IF(TEST0.GE.0.0)GO TO 888
  800 DO 13 I=1,2
      IFLAG(I)=0
      XINT(I)=0.0
      YINT(I)=0.0
   13 ZINT(I)=0.0
      GO TO 999
C*****SOLVE QUADRATIC FOR X,Y OR Z DEPENDING ON CASE
  888 GO TO (100,200,300),ICASE
  100 XINT(1)=QROOT(+1,A,B,C)
      XINT(2)=QROOT(-1,A,B,C)
      DO 11 I=1,2
      YINT(I)=YI+SY*XINT(I)
   11 ZINT(I)=ZI+SZ*XINT(I)
      GOTO 400
  200 YINT(1)=QROOT(+1,A,B,C)
      YINT(2)=QROOT(-1,A,B,C)
      DO 21 I=1,2
      XINT(I)=XI+SX*YINT(I)
   21 ZINT(I)=ZI+SZ*YINT(I)
      GO TO 400
  300 ZINT(1)=QROOT(+1,A,B,C)
      ZINT(2)=QROOT(-1,A,B,C)
      DO 31 I=1,2
      XINT(I)=XI+SX*ZINT(I)
   31 YINT(I)=YI+SY*ZINT(I)
C*** TEST FOR VALID INTERCEPTS
400   I1=0
      I2=0
      IFLAG(1)=2
      IFLAG(2)=2
      IF (ZINT(1).GE.0..AND.XINT(1).GE.0..AND.XINT(1).LE.X0) I1=1
      IF (ZINT(2).GE.0..AND.XINT(2).GE.0..AND.XINT(2).LE.X0) I2=1
      IF (I1.EQ.0.OR.I2.EQ.0) GOTO 450
      IF (XINT(1).EQ.XINT(2).AND.YINT(1).EQ.YINT(2).AND.ZINT(1).EQ.
     *    ZINT(2)) I2=0
      IF (I1.EQ.1.AND.I2.EQ.1) GOTO 600
C*** AT LEAST ONE INTERCEPT INVALID. FIRST COMPUTE POSSIBLE Z=0. INTCPT
450   GOTO (460,470,480),ICASE
460   Z=0.
      IF (SZ.EQ.0.) GOTO 500
      X=-ZI/SZ
```

```
      Y=YI+SY*X
      GOTO 490
470   Z=0.
      IF (SZ.EQ.0.) GOTO 500
      Y=-ZI/SZ
      X=XI+SX*Y
      GOTO 490
480   Z=0.
      X=XI
      Y=YI
C*** TEST IF Z=0. INTERCEPT VALID.
490   IF (X.GT.X0.OR.X.LT.0.) GOTO 500
      IF (NCODE.EQ.1) TEST3=(Y/Y0)**2-(X/X0)**2
      IF (NCODE.EQ.2) TEST3=((X-X0)/X0)**2 + (Y/Y0)**2 - 1.
      IF (TEST3.GT.0.) GOTO 500
      IF (I1.EQ.1) GOTO 495
      XINT(1)=X
      YINT(1)=Y
      ZINT(1)=Z
      I1=1
      IF (I2.EQ.1) GOTO 600
      GOTO 500
495   XINT(2)=X
      YINT(2)=Y
      ZINT(2)=Z
      I2=1
      IF (I1.EQ.1) GOTO 600
C*** NEXT COMPUTE INTERCEPT ON X=X0
500   GOTO (510,505,507),ICASE
505   X=X0
      Y=(X-XI)/SX
      Z=ZI+SZ*Y
      GOTO 515
507   X=X0
      Z=(X-XI)/SX
      Y=YI+SY*Z
      GOTO 515
510   X=X0
      Y=YI+SY*X
      Z=ZI+SZ*X
515   IF (Z.LT.0.) GOTO 800
      TEST4=(Y/Y0)**2 + (Z/Z0)**2 - 1.
      IF (TEST4.GT.0.) GOTO 800
      IF (I1.EQ.1) GOTO 520
      XINT(1)=X
      YINT(1)=Y
      ZINT(1)=Z
      I1=1
      IF (I2.EQ.0) GOTO 800
      GOTO 600
520   XINT(2)=X
      YINT(2)=Y
      ZINT(2)=Z
      I2=1
600   DO 620 I=1,2
      X1=(XL(1)-XINT(I))**2 + (YL(1)-YINT(I))**2 + (ZL(1)-ZINT(I))**2
      X2=(XL(2)-XINT(I))**2 + (YL(2)-YINT(I))**2 + (ZL(2)-ZINT(I))**2
      DIST(I)=SQRT(X1)+SQRT(X2)
620   CONTINUE
      TEST5=1.0001*SQRT(DELX**2+DELY**2+DELZ**2)
      IF(DIST(1).GT.TEST5.AND.DIST(2).GT.TEST5) GOTO 800
      IF(INT.EQ.0) GOTO 700
C*** REPLACE ONE INTERCEPT WITH INTERIOR OBS/TGT.
      DO 610 I=1,2
      IF (XL(INT).GT.(XINT(I)+.001).OR.XL(INT).LT.(XINT(I)-.001))
     * GOTO 610
      IF (YL(INT).GT.(YINT(I)+.001).OR.YL(INT).LT.(YINT(I)-.001))
     * GOTO 610
      IF (ZL(INT).GT.(ZINT(I)+.001).OR.ZL(INT).LT.(ZINT(I)-.001))
     *  GOTO 610
```

```
      IFLAG(I)=1                                                        XYZ02810
      IF (LEAD.EQ.INT) IFLAG(I)=3                                       XYZ02820
      IF (ISURF.EQ.INT) IFLAG(I)=2                                      XYZ02830
      GOTO 700                                                          XYZ02840
610   CONTINUE                                                          XYZ02850
C*** REPLACE CLOSEST INTERCEPT OUTSIDE LOS RANGE                        XYZ02860
      K1=1                                                              XYZ02870
      IF (DIST(1).LT.DIST(2)) K1=2                                      XYZ02880
      XINT(K1)=XL(INT)                                                  XYZ02890
      YINT(K1)=YL(INT)                                                  XYZ02900
      ZINT(K1)=ZL(INT)                                                  XYZ02910
      IFLAG(K1)=1                                                       XYZ02920
700   IF (XINT(1).GE.(X0-.001).AND.XINT(1).LE.(X0+.001)) IFLAG(1)=3     XYZ02930
      IF (XINT(2).GE.(X0-.001).AND.XINT(2).LE.(X0+.001)) IFLAG(2)=3     XYZ02940
 999  RETURN                                                            XYZ02950
      END                                                               XYZ02960
```

```
      SUBROUTINE BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,
     *BRAT1,BRAT2,BRAT3,BRAT4,BRAT5)
C
C     THIS ROUTINE PROVIDES DEFAULT MUNITION CHARACTERISTIC VALUES TO
C     SMOKE.  SMOKE MUNITION TYPES (TYPM) ARE GIVEN IN THE COMMENTS
C     OF THE MAIN ROUTINE IN SMOKE
C
      DIMENSION B1(21),B2(21),B3(21),B4(21),B5(21)
      DIMENSION F(21),T(21),E(21),IT(21)
      DATA B1 / .537,.631,.2218,.537,.2218,1.,1.,1.,1.,1.,1.,0.,
     *      0.,.521,1.631,1.808,.1204,.653,1.731,0.,1. /
      DATA B2 /.476,-.4985,3.915,.476,3.915,0.,0.,0.,0.,0.,0.,  0.,  0.,
     *           2.106,.678,-2.556,3.1012,-3.136,-2.852,3.6832,0./
      DATA B3 / 4.779,6.745,-1.7368,4.779,-1.7368,0.,0.,0.,0.,0.,0.,
     * 0., 0.,   -1.11,-5.907,2.883,-2.2104,15.309,4.341,-5.3472,0./
      DATA B4 / -5.472,-6.52,-2.3995,-5.472,-2.3995,0.,0.,0.,0.,0.,0.,
     * 0., 0.,   -.748,4.012,-2.008,.206,-12.872,-3.108,3.8348,0./
      DATA B5 / 11*0.,2*1.,8*0./
      DATA IT /3,3,3,3,3,3,1,1,1,1,1,2,2,2,2,2,2,5,5,5,4 /
      DATA  T /100., 70.,120.,100.,120.,900.,1.,1.,1.,1.,1.,600.,
     *     600.,240.,470.,390.,721.,260.,380.,750.,900. /
C
      DATA  F /5.46,2.69,1.65,17.19,7.50,30.,0.76,1.75,8.14,15.6,
     *           3.83,8.14,13.52,.463,.139,.234,19.98,.128,.0243,
     *           19.4,40.0 /
      DATA E / 40.,40.,40.,40.,40.,24.,100.,100.,100.,100.,100.,
     *          60.,60.,66.,71.,67.,77.,53.,55.,51.,100. /
      MAXS=21
      ITP=IFIX(TYPM+.0001)
      IF (ITP.EQ.0) RETURN
      IF (ITP.LE.MAXS) GOTO 10
      IERR=1
      RETURN
10    BRAT1=B1(ITP)
      BRAT2=B2(ITP)
      BRAT3=B3(ITP)
      BRAT4=B4(ITP)
      BRAT5=B5(ITP)
      ITYPE=IT(ITP)
      IF (MUNRD.NE.0) GOTO 20
      XN=1.
      YF=0.
      TBURN=T(ITP)
      EFF=E(ITP)
      FW=F(ITP)
      RETURN
20    IF (XN.EQ.0.) XN=1.
      IF (TBURN.EQ.0.) TBURN=T(ITP)
      IF (EFF.EQ.0.) EFF=E(ITP)
      IF (FW.EQ.0.) FW=F(ITP)
      RETURN
      END
```

```
      SUBROUTINE GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1,
     *ZPP2)
C
C  THIS ROUTINE COMPUTES THE CL CONTRIBUTION FROM AN EXOTHERMIC,
C     BUOYANTLY RISING SMOKE CLOUD OF GAUSSIAN DISTRIBUTION AND
C     UNIT CONCENTRATION.  IT IS CENTERED ON THE LEADING EDGE AT
C     COORDINATES  (A,B,C-2*SIGZ) WHERE
C INPUTS    A    = CLOUD DOWNWIND DISTANCE
C           B    = CLOUD BASE HALF-WIDTH (AT LEADING EDGE)
C           C    = CLOUD HEIGHT (AT LEADING EDGE)
C           C2   = MOMENTUM RISE COEFFICIENT. (USUALLY BRIGGS OR LIMITED
C                  RISE BRIGGS)
C           XPP1,...ZPP2 COORDINATES OF TARGET AND OBSERVER IN MUNITION
C                  CENTERED COORDINATE SYSTEM WITH WIND VECTOR X-AXIS
C OUTPUT    CLGAUS = CL VALUE FOR UNIT MASS (METER**-2)
C           KCALL  = SET TO 1 AFTER EVERY CHANGE IN TGT/OBS COORD.
      DIMENSION AV(2),AP(2),BP(2),CP(2)
      CLGAUS=0.
      IF (KCALL.NE.0) GOTO 100
      KCALL=1
      ICASE=0
C*** COMPUTE LOS GENERALIZED COORDINATES.
      IF (ZPP1.LT.0. .AND. ZPP2.LT.0.) RETURN
      DELX=XPP2-XPP1
      DELY=YPP2-YPP1
      DELZ=ZPP2-ZPP1
      IF (ABS(DELX).LE..01*ABS(DELY) .OR. ABS(DELX).LE..01*ABS(DELZ))
     1    GOTO 10
      ICASE=1
      SX=DELX/DELX
      SY=DELY/DELX
      SZ=DELZ/DELX
      XI=XPP1-SX*XPP1
      YI=YPP1-SY*XPP1
      ZI=ZPP1-SZ*XPP1
      AV(1)=XPP1
      AV(2)=XPP2
      GOTO 80
10    IF (ABS(DELY).LE..01*ABS(DELZ)) GOTO 20
      ICASE=2
      SX=DELX/DELY
      SY=DELY/DELY
      SZ=DELZ/DELY
      XI=XPP1-SX*YPP1
      YI=YPP1-SY*YPP1
      ZI=ZPP1-SZ*YPP1
      AV(1)=YPP1
      AV(2)=YPP2
      GOTO 80
20    IF (ABS(DELZ).LT..001) RETURN
      ICASE=3
      SX=DELX/DELZ
      SY=DELY/DELZ
      SZ=DELZ/DELZ
      XI=XPP1-SX*ZPP1
      YI=YPP1-SY*ZPP1
      ZI=ZPP1-SZ*ZPP1
      AV(1)=ZPP1
      AV(2)=ZPP2
80    IF (ZPP1.LT.0.) AV(1)=-ZI/SZ
      IF (ZPP2.LT.0.) AV(2)=-ZI/SZ
      SMUL=SQRT(SX*SX+SY*SY+SZ*SZ)
100   IF (ICASE.EQ.0) RETURN
C*** COMPUTE GAUSSIAN PARAMETERS, REAL AND REFLECTED IMAGE CLOUD TO
C    ACCOUNT FOR GROUND REFLECTED SMOKE
      SIGZ=(2.73+C2*A)/2.15
      ZB=C-(2.73+C2*A)
      IF (ZB.GE.C) RETURN
      SIGY=B*SQRT(1.-(ZB/C)**2)/2.15
      IF (ZB.LT.0.) SIGY=B/2.15
```

```
      SIGX=SIGY
      ASIG=(SX/SIGX)**2 + (SY/SIGY)**2 + (SZ/SIGZ)**2
      BMEAN=2.*((SX*(XI-A)/SIGX**2) + (SY*YI/SIGY**2))
      BP(1)=BMEAN+2.*SZ*(ZI-ZB)/SIGZ**2
      BP(2)=BMEAN+2.*SZ*(ZI+ZB)/SIGZ**2
      CTOT=((XI-A)/SIGX)**2 + (YI/SIGY)**2
      CP(1)=CTOT + ((ZI-ZB)/SIGZ)**2
      CP(2)=CTOT + ((ZI+ZB)/SIGZ)**2
C*** CALCULATE FOR LOS INTEGRAL
      CMUL=SMUL/(2.*3.14159*SIGX*SIGY*SIGZ*SQRT(ASIG))
      DO 220 I=1,2
      CEXU=.5*(CP(I)-(BP(I)**2)/(4.*ASIG))
      IF (CEXU.GT.20.) GOTO 220
C***  INFINITE PATH LOS
      CLU=EXP(-CEXU)
C***  CORRECTION FOR FINITE PATH
      DO 210 J=1,2
      AP1=(AV(J)+BP(I)/(2.*ASIG))*SQRT(ASIG/2.)
      P1=ABS(AP1)
      CP1=0.
      IF (P1.LE.5.) CP1=0.5/(1.+P1*(.0705230784+P1*(.042282013+P1*(
     *.0092705272+P1*(.0001520134+P1*(.0002765672+P1*.0000430638))))))
     ***16
      IF (AP1.GE.0.) CP1=1.-CP1
      AP(J)=CP1
210   CONTINUE
      CLGAUS=CLGAUS+CLU*ABS(AP(2)-AP(1))
220   CONTINUE
      CLGAUS=CMUL*CLGAUS
      RETURN
      END
```

```
      SUBROUTINE LZTRAN(WAVE1,ICLMAT,LAZTRN,IERR)
C
C  THIS SUBROUTINE CALCULATES MOLECULAR ABSORPTION COEFFICIENTS AT
C  LASER FREQUENCIES. PH20 AND T ARE THE PARTIAL PRESSURE OF WATER
C  VAPOR AND TEMPERATURE, IN TORR AND DEGREES K RESPECTIVELY.
C  LID IS THE LASER LINE IDENTIFICATION AS DESCRIBED BELOW.
C  ABCOEF IS THE ABSORPTION COEFFICIENT RETURNED IN KM-1.
C
C*************************************************************
C           INPUT
C  CARD 1     LNAME1, LNAME2, PH20, T, LZPATH FORMAT (2A4,3F10.3)
C
C      LNAME1  FIRST 4 CHARS OF LASER LINE   (A4)
C      LNAME2  SECOND 4 CHARS OF LASER LINE   (A4)
C           ***IF LNAME NOT ENTERED WILL USE WAVELENGTH READ IN MAIN
C
C      PH20    WATER VAPOR  PRESSURE  0. TO .35   (MB)  (F10.3)
C
C      T       AMBIENT AIR TEMPERATURE 260. TO .320  (C)  (F10.3)
C
C      LZPATH        PATHLENGTH IN KM F(F10.3)
C           ***PH20, AND T NOT REQUIRED WHEN ICLIMATE(IN MAIN)=1 OR 2
C*************************************************************
C           OUTPUT
C      LZTRAN    TRANSMISSION
C
C      NOTES
C     ABCOEF  RETURNS THE ABSORPTION COEFFICIENT (KM-1)
C
C     LNAME1  FIRST 4 CHARS OF LASER LINE ON NORMAL RETURN
C                                 BLANK ON ERROR RETURN
C
C     LNAME2  SECOND 4 CHARS OF LASER LINE ON NORMAL RETURN
C                                  BLANK ON ERROR RETURN
C
C      ++ CALLED PROGRAMS ++
C
C     LZIDNM
C
C**********************************************
C LASER LINE IDENTIFICATION
C  LID=1  ND:YAG LASER, 1.06 MICRONS
C  LID=2 CO2 LASER LINE P(20) ,  10.591 MICRONS
C  LID=101 TO 127 OF LASER, 3.521 TO 4.089 MICRONS
C   101 P3(12) * 107 P3(8)  * 113 P3(5)  * 119 P2(5)  * 125 P1(5)
C   102 P3(11) * 108 P2(11) * 114 P2(8)  * 120 P1(8)  * 126 P1(4)
C   103 P3(10) * 109 P3(7)  * 115 P2(7)  * 121 P2(4)  * 127 P1(3)
C   104 P2(13) * 110 P2(10) * 116 P1(10) * 122 P1(7)  *
C   105 P3(9)  * 111 P3(6)  * 117 P2(6)  * 123 P2(3)  *
C   106 P2(12) * 112 P2(9)  * 118 P1(9)  * 124 P1(6)  *
C  LID=201 TO 219 CO LASER, 4.908 TO 5.088 MICRONS
C   201 P6(12) * 205 P6(8)  * 209 P5(12)  * 213 P5(8)  * 217 P4(9)
C   202 P6(11) * 206 P5(15) * 210 P5(11)  * 214 P5(7)  * 218 P4(8)
C   203 P6(10) * 207 P5(14) * 211 P5(10)  * 215 P4(11) * 219 P4(7)
C   204 P6(9)  * 208 P5(13) * 212 P5(9)   * 216 P4(10) *
C  LID=301 TO 305  GA AS LASER, (GA.85 TO GA.950) LASER LINE NAMES
C   301 0.850 MICROMETERS      *  304  0.925 MICROMETERS
C   302 0.875 MICROMETERS      *  305  0.950 MICROMETERS
C   303 0.900 MICROMETERS      *
      INTEGER LNAME1,LNAME2,BLANK,LNAME3,LNAME4
      REAL LAZTRN,LZPATH
      DIMENSION ADF0(30),ADF1(30),ADF2(30),ADF3(30),ADF4(30)
     1          ,ADF5(30)
      DIMENSION ACO0(20),ACO1(20),ACO2(20),ACO3(20),ACO4(20)
     1          ,ACO5(20)
      DIMENSION AGA0(5),AGA1(5),AGA2(5),AGA3(5),AGA4(5),
     1          AGA5(5),AGA6(5),AGA7(5),AGA8(5)
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
     1       FOGPRB,WNDVEL,WNDDIR,IPASCT
```

```
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /GEOMET/PTS(15),IGEOSW
C      THE POLYNOMIAL COEFFICIENTS ARE SELECTED BY THE LID.  THE
C    INDEX FOR THE COEFFICIENT ARRAYS FOR THE DF LASER IS
C    I = LID - 100.  NOW I IS IN THE RANGE 1..27 SINCE THERE ARE
C    27 DF LASER LINES.  WHEN THE POLYNOMIAL IS EVALUATED I IS
C    USED TO INDEX THE ARRAYS ADF/0..5/ THUS SELECTING THE
C    CORRECT COEFFICIENTS FOR THE LASER LINE SELECTED.
C    COEFFICIENTS FOR THE OTHER LASER POLYNOMIALS ARE SELECTED
C    IN THE SAME FASHION.
C    POLYNOMIAL COEFFICIENTS  FOR DF LASER LINES, (1..27)
      DATA ADF0/.1019,.08352,.04083,.03675,.02042,.01833,
     1     .04738,.03134,.07870,.05844,.1096E-2,.8353E-
     2     2,.2537E-2,.3254E-2,-.1103E-2,-.6471E-3,-.1423E-
     3     3,-.4664E-2,-.1221E-4,-.1698E-3,.1172E-3,.6195E-
     4     2,.1272E-2,.5485E-2,.1651E-2,.6913E-2,-.4498E-
     5     2,3*0./
      DATA ADF1/-.9718E-4,-.1160E-3,-.4892E-4,-.4230E-4,-
     1     .1750E-4,-.1524E-4,-.5589E-4,-.4642E-4,-.1218E-
     2     3,-.3683E-4,.1672E-4,-.1346E-4,-.4765E-6,-.8548E-
     3     6,.6089E-5,.8897E-5,.1507E-5,.4448E-4,.4540E-
     4     7,.5569E-6,.1327E-6,-.1462E-4,-.7524E-6,-.1398E-
     5     5,-.7025E-6,-.1044E-4,.1816E-4,3*0./
      DATA ADF2/.9666E-2,.7252E-2,.7050E-2,.7142E-2,.6320E-
     1     2,.6191E-2,.5400E-2,.5344E-2,.4064E-2,.4682E-
     2     2,.3734E-2,.5839E-2,.5075E-2,.2484E-2,.8190E-
     3     2,.6920E-2,.8779E-2,.7914E-2,.7094E-2,.5327E-
     4     3,.6692E-2,.9452E-2,.01025,.01367,.01279,.7844E-
     5     2,.01201,3*0./
      DATA ADF3/-.2655E-4,-.1805E-4,-.1879E-4,-.1937E-4,-
     1     .1704E-4,-.1669E-4,-.1412E-4,-.1417E-4,-.9076E-
     2     5,-.1165E-4,-.7371E-5,-.9113E-5,-.1290E-4,-
     3     .1070E-5,-.6746E-5,-.1573E-4,-.1683E-4,-.1883E-
     4     4,-.1774E-4,.3002E-4,-.1025E-4,-.2488E-4,-.2596E-
     5     4,-.2848E-4,-.2858E-4,-.4991E-5,-.2608E-4,3*0./
      DATA ADF4/.7847E-4,.5729E-4,.5585E-4,.5606E-4,.4847E-
     1     4,.4668E-4,.4145E-4,.4218E-4,.3314E-4,.3642E-
     2     4,.3170E-4,.3682E-4,.3835E-4,.4798E-4,.3537E-
     3     4,.4635E-4,.4941E-4,.5601E-4,.5494E-4,-.1120E-
     4     4,.6567E-4,.9104E-4,.7398E-4,.8509E-4,.8746E-
     5     4,.1050E-3,.1060E-3,3*0./
      DATA ADF5/-.2056E-6,-.1374E-6,-.1408E-6,-.1432E-6,-
     1     .1222E-6,-.1172E-6,-.1011E-6,-.1054E-6,-.7169E-
     2     7,-.8600E-7,-.6588E-7,-.8998E-7,-.9081E-7,-
     3     .1320E-6,-.1570E-7,-.9751E-7,-.8159E-7,-.1234E-
     4     6,-.1313E-6,.1836E-6,-.1469E-6,-.2540E-6,-.1796E-
     5     6,-.1691E-6,-.1886E-6,-.2635E-6,-.2855E-6,3*0./
C    POLYNOMIAL COEFFICIENTS FOR CO LASER LINES, (1..19)
      DATA AC00/-1.813E-3,-9.289E-4,1.153E-3,-1.985E-3,-
     1     4.523E-3,-1.205E-3,-2.225E-4,-4.061E-3,-4.522E-
     2     2,-2.267E-6,-5.917E-3,-1.423E-3,-3.640E-3,1.096E-
     3     3,6.455E-4,-3.922E-3,-1.873E-5,-1.055E-4,1.489E-
     4     2,0./
      DATA AC01/3 426E-5,3.658E-6,-1.372E-6,7.229E-6,1.641E-
     1     5,6.428E-6,1.042E-7,1.435E-5,1.593E-4,-5.334E-
     2     7,1.498E-5,5.284E-6,1.806E-5,-3.651E-6,-1.303E-
     3     6,1.835E-5,4.755E-6,2.330E-6,6.196E-6,0./
      DATA AC02/8.813E-2,-1.020E-1,4.881E-2,6.872E-2,-
     1     6.244E-1,4.474E-2,1.226E-2,-1.462E-2,1.490E-
     2     1,1.428E-2,1.934,9.034E-3,-1.091E-1,1.284E-2,-
     3     2.131E-2,6.543E-3,2.824E-2,9.463E-3,-9.885E-2,0./
      DATA AC03/8.384E-4,1.211E-3,-4.687E-6,-7.765E-
     1     5,2.641E-3,-3.135E-5,1.620E-4,6.707E-4,2.581E-
     2     3,4.018E-4,-3.113E-3,9.692E-5,7.907E-4,1.070E-
     3     4,3.026E-4,1.216E-4,-1.859E-5,4.274E-5,6.241E-
     4     4,0./
      DATA AC04/2.850E-4,-4.934E-5,-2.176E-5,1.253E-
     1     4,1.158E-3,1.926E-5,9.823E-5,-2.395E-4,-6.255E-
     2     5,-6.630E-5,4.851E-3,-5.183E-5,-6.993E-4,-8.196E-
     3     5,-2.239E-4,-1.120E-4,5.415E-5,-8.393E-5,-5.893E-
```

```
     4      4,0./
      DATA AC05/4.209E-6,3.426E-6,5.383E-7,6.252E-8,-4.149E-
     1      6,2.374E-7,-8.637E-9,3.454E-6,1.526E-5,2.140E-6,-
     2      4.282E-6,5.433E-7,4.231E-6,7.778E-7,1.695E-
     3      6,8.830E-7,-1.548E-8,4.629E-7,3.241E-6,0./
C  POLYNOMIAL COEFFICIENTS FOR CO2 LASER
      DATA AC020,AC021,AC022,AC023,AC024,AC025,AC026,AC027,
     1      AC028/.4488,-4.1864E-3,5.7903E-2,-3.6770E-
     2      4,3.8521E-3,-4.7330E-6,1.0000E-5,6.0131E-7,-
     3      1.7441E-8/
C  POLYNOMIAL COEFFICIENTS FOR GA AS LASER, (1..5)
      DATA AGA0/7.947E-3,7.590E-3,1.010,0.6094,4.271/
      DATA AGA1/-3.543E-5,-3.544E-5,-8.012E-3,-4.777E-3,-
     1      3.371E-2/
      DATA AGA2/1.740E-4,1.203E-4,8.736E-2,4.527E-2,.3364/
      DATA AGA3/-5.855E-7,-5.093E-7,-2.135E-4,-1.104E-4,-
     1      8.425E-4/
      DATA AGA4/-1.282E-5,-9.236E-6,-3.964E-3,-2.154E-3,-
     1      1.615E-2/
      DATA AGA5/5.586E-8,4.720E-8,1.596E-5,8.512E-6,6.350E-
     1      5/
      DATA AGA6/5.124E-8,5.235E-8,1.600E-5,9.392E-6,6.705E-
     1      5/
      DATA AGA7/2.706E-10,4.379E-10,-1.396E-7,-6.982E-8,-
     1      4.721E-7/
      DATA AGA8/-4.963E-11,-5.728E-11,-1.125E-8,-5.657E-9,-
     1      4.113E-8/
      DATA BLANK/1H /
C     CHANGE ACCURACY TO 3 DECIMAL PTS (PGM DATA LIMIT)
      WAVEL=FLOAT(IFIX(1000.*WAVE1))/1000.
      READ (IOIN,1100) LNAME1,LNAME3,LNAME2,LNAME4,PH20,T,LZPATH
      IF(IGEOSW.NE.1)GO TO 99
      LZPATH=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+
     +(PTS(6)-PTS(3))**2)
   99 CONTINUE
C     CHANGE UNITS - MB TO TORR; C TO K
      PH20=PH20/TORRMB
      T=T+CDEGK
      IF (ICLMAT.EQ.1) T=TEMP+CDEGK
      IF (ICLMAT.EQ.1) PH20=6.11*10.**
     1  (7.5*TEMP/(TEMP+237.3))*RH/(100.*TORRMB)
C  PRINT HEADER WHEN THE WAVELENGTH CHANGES
      IF (OLDWAV.NE.WAVEL) WRITE (IOOUT,1000)
      OLDWAV=WAVEL
      ABCOEF=0.
      IF (WAVEL.EQ.0.0) GO TO 100
      IF (WAVEL.LT.0.8.OR.WAVEL.GT.11.0) GO TO 900
  100 CALL LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID)
      IF (LID.EQ.0) RETURN
      P2=PH20*PH20
      IF (T.GE.260.AND.T.LE.320.AND.PH20.GE.0.AND.
     1     PH20.LE.35) GO TO 200
C  PRINT WARNING THAT TEMP OR PRESSURE IS OUT OF RANGE FOR
C  ACCURATE CALCULATIONS AND CONTINUE.
      WRITE (IOOUT,1300)
  200 IF (LID.GT.100) GO TO 500
      IF (LID.LT.1.OR.LID.GT.2) GO TO 900
      IF (LID-2) 300,400,900
C   ND:YAG LASER. NO MOLECULAR ABSORPTION AT 1.06 MICRONS.
  300 GO TO 800
C  CO2 LASER LINE P(20)
  400 T2=T*T
      ABCOEF=AC020+AC021*T+AC022*PH20+AC023*T*PH20+AC024*P2+
     1      AC025*T*P2+AC026*T2+AC027*T2*PH20+AC028*T2*P2
      GO TO 800
  500 IF (LID.GT.200) GO TO 600
C  DF LASER.  I IS THE LASER LINE INDEX
      I=LID-100
      IF (I.GT.27) GO TO 900
      ABCOEF=ADF0(I)+ADF1(I)*T+ADF2(I)*PH20+ADF3(I)*T*PH20+
```

```
     1         ADF4(I)*P2+ADF5(I)*T*P2
      GO TO 800
  600 IF (LID.GT.300) GO TO 700
C   CO LASER.  I IS THE LASER LINE INDEX.
      I=LID-200
      IF (I.GT.19) GO TO 900
      ABCOEF=ACO0(I)+ACO1(I)*T+ACO2(I)*PH2O+ACO3(I)*T*PH2O+
     1         ACO4(I)*P2+ACO5(I)*T*P2
      GO TO 800
  700 IF (LID.GT.400) GO TO 900
C   GA AS LASER.  I IS WAVELENGTH INDEX.
      I=LID-300
      IF (I.GT.5) GO TO 900
      T2=T*T
      ABCOEF=AGA0(I)+AGA1(I)*T+AGA2(I)*PH2O+AGA3(I)*T*PH2O+
     1         AGA4(I)*P2+AGA5(I)*T*P2+AGA6(I)*T2+AGA7(I)*T2*
     2         PH2O+AGA8(I)*T2*P2
      GO TO 800
C   NORMAL RETURN
  800 IF (ABCOEF.LT.0.) ABCOEF=0.
C     COMPUTE TRANSMISSION
      LAZTRN=EXP(-LZPATH*ABCOEF)
      WRITE (IOOUT,1200) WAVEL,PH2O,T,ABCOEF,LNAME1,LNAME3,
     +LNAME2,LNAME4,LZPATH,LAZTRN
      RETURN
C   ERROR RETURN
  900 WRITE (IOOUT,1400)
      LNAME1=BLANK
      LNAME3=BLANK
      LNAME2=BLANK
      LNAME4=BLANK
      LAZTRN=1.
      IERR=1
      RETURN
C
 1000 FORMAT (/,69X,10HABSORPTION,/,23X,11H WAVELENGTH,4X,
     1          12HH2O PRESSURE,4X,11HTEMPERATURE,4X,
     2          11HCOEFFICIENT,6X,4HLINE,9X,10HPATHLENGTH,4X,
     3          12HTRANSMISSION,/,24X,9H(MICRONS),8X,6H(TORR)
     4          ,11X,5H(ABS),10X,6H(KM-1),24X,4H(KM),/)
 1100 FORMAT (4(A2),3F10.3)
 1200 FORMAT (1H ,22X,F09.3,F15.3,F16.2,E16.3,7X,4(A2),5X,
     1          E10.4,5X,E10.4)
 1300 FORMAT (39H *** WARNING VALUE OF T OR PH2O OUT OF ,
     1          10HRANGE *** /28H     T RANGE = 260 TO 320 K   ,
     2          25HPH2O RANGE = 0 TO 35 TORR)
 1400 FORMAT (40H *** ERROR WAVELENGTH OUT OF ACCEPTABLE
     1          7HRANGE: ,26H .8 TO 11.0 MICRONS ***    ,/,
     2          37H CONTROL RETURNED TO MAIN FROM LZTRAN)
      END
```

```
      SUBROUTINE LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID)
C*****************************************************
C
C     THIS SUBROUTINE CONVERTS THE WAVELENGTH IN MICRONS OR THE LASER
C   LINE NAME, IF WAVEL = 0, TO AN INTEGER LASER ID NUMBER WHICH IS
C   USED BY LZTRAN.
C      ++ WAVEL TO LID, WHEN WAVEL NOT = 0  ++
C   THE SUBROUTINE DOES A BINARY SEARCH OF THE ARRAY AWAVEL TO FIND
C   A MATCH, WAVEL = AWAVEL(K).  WHEN A MATCH IS FOUND THE LASER ID
C   IS LOADED FROM THE ID ARRAY, LID = AID(K).  THE LASER LINE NAME
C   IS ALSO LOADED INTO TWO VARIABLES, LNAME1 = INAME1(K) LNAME2 =
C   INAME2(K).  IF AN EXACT MATCH IS NOT FOUND THE CLOSEST STANDARD
C   TO THE PARAMETER WAVEL IS USED.  A WARNING IS PRINTED AND K IS
C   SET TO THE PROPER VALUE SO IT CAN BE USED TO INDEX THE ID AND
C   LINE ARRAYS.
C      ++ LASER LINE TO LID, WHEN WAVEL = 0  ++
C   WHEN THE WAVELENGTH PARAMETER IS ZERO THE CONVERSION IS DONE
C   FROM LASER LINE NAME TO LID.  A SEQUENTIAL SEARCH OF THE LINE
C   NAME ARRAY IS PERFORMED.  WHEN A MATCH IS FOUND K IS SET AND
C   WAVEL AND LID ARE LOADED FROM THE APPROPRIATE ARRAYS.  WHEN
C   NO MATCH IS FOUND AN ERROR MESSAGE IS PRINTED AND LID IS SET
C   TO ZERO.  LID IS USED TO NOTIFY LZTRAN THAT AN ERROR HAS
C   OCCURED AND NO CALCULATIONS SHOULD BE PERFORMED.
C
C      ++   PARAMETERS ++
C
C     WAVEL    LASER WAVELENGTH (MICRONS)
C
C        *** INPUTS IF WAVEL = 0.0 ***
C     LNAME1  FIRST 4 CHARS OF LASER LINE
C
C     LNAME2  SECOND 4 CHARS OF LASER LINE
C
C      ++ RESULTS ++
C
C     LID      LASER LINE IDENTIFIER
C
C     LNAME1  FIRST 4 CHARS OF LASER LINE
C
C     LNAME2  SECOND 4 CHARS OF LASER LINE
C
C*****************************************************
      INTEGER AID(53)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DIMENSION AWAVEL(53),INAME1(106),INAME2(53)
      DATA AWAVEL/.85,.875,.9,.
     1     925,.95,1.06,3.521,3.55,3.581,3.612,3.636,3.645,
     2     3.666,3.679,3.698,3.715,3.731,3.752,3.765,3.8,
     3     3.82,3.837,3.854,3.875,3.89,3.915,3.927,3.956,
     4     3.965,3.999,4.005,4.046,4.089,4.908,4.918,4.928,
     5     4.938,4.948,4.972,4 982,4.992,5.002,5.012,5.022,
     6     5.032,5.043,5.047,5.054,5.057,5.067,5.078,5.088,
     7     10.591/
      DATA AID/301,302,303,304,305,1,127,126,125,124,123,
     1     122,121,120,119,118,117,116,115,114,113,112,111,
     2     110,109,108,107,106,105,104,103,102,101,219,218,
     3     217,216,215,214,213,212,211,210,209,208,207,205,
     4     206,204,203,202,201,2/
      DATA INAME1/2HGA,2H.8,2HGA,2H.8,2HGA,2H.9,2HGA,2H.9,2HGA,2H.9,
     1 2HRU,2HBY,2HP1,2H(3,2HP1,2H(4,2HP1,2H(5,2HP1,2H(6,2HP2,2H(3,
     2 2HP1,2H(7,2HP2,2H(4,2HP1,2H(8,2HP2,2H(5,2HP1,2H(9,2HP2,2H(6,
     3 2HP1,2H(1,2HP2,2H(7,2HP2,2H(8,2HP3,2H(5,2HP2,2H(9,2HP3,2H(6,
     4 2HP2,2H(1,2HP3,2H(7,2HP2,2H(1,2HP3,2H(8,2HP2,2H(1,2HP3,2H(9,
     5 2HP2,2H(1,2HP3,2H(1,2HP3,2H(1,2HP3,2H(1,2HP4,2H(7,2HP4,2H(8,
     6 2HP4,2H(9,2HP4,2H(1,2HP4,2H(1,2HP5,2H(7,2HP5,2H(8,2HP5,2H(9,
     7 2HP5,2H(1,2HP5,2H(1,2HP5,2H(1,2HP5,2H(1,2HP5,2H(1,2HP6,2H(8,
     8 2HP5,2H(1,2HP6,2H(9,2HP6,2H(1,2HP6,2H(1,2HP6,2H(1,2HP(,2H20/
      DATA INAME2/1H5,2H75,1H ,2H25,1H5,1H ,11*1H),2H0),5*
     1     1H),2H0),1H),2H1),1H),2H2),1H),2H3),2H0),2H1),
     2     2H2),3*1H),2H0),2H1),3*1H),2H0),2H1),2H2),2H3),
```

```
     3      2H4),1H),2H5),1H),2H0),2H1),2H2),1H)/
      DATA IFIRST,ILAST/1,53/
      I=IFIRST
      J=ILAST
      TWAVEL=WAVEL
C   CHECK FOR WAVELENGTH OR LASER LINE PASSED AS INPUT.  IF
C   WAVEL = 0 THEN DO A SEQUENTIAL SEARCH ON LINE NAME,
C   LNAME1,LNAME2.
      IF (WAVEL.NE.0.0) GO TO 200
C   INPUT = LASER LINE NAME
C   SEQUENTIAL SEARCH LOOP
      DO 100 K=IFIRST,ILAST
      KK=2*K-1
      IF((LNAME1.EQ.INAME1(KK)).AND.(LNAME3.EQ.INAME1(KK+1))) GO TO 10
      GO TO 100
   10 IF (LNAME2.EQ.INAME2(K)) GO TO 600
  100 CONTINUE
C   ERROR, NO MATCH ON LASER NAME
      LID=0
C   PRINT ERROR MESSAGE
      WRITE (IOOUT,900) LNAME1,LNAME3,LNAME2,LNAME4
      GO TO 700
C   INPUT = WAVELENGTH
C   BINARY SEARCH LOOP
  200 K=(I+J)/2
      IF (WAVEL.LE.AWAVEL(K)) J=K-1
      IF (WAVEL.GE.AWAVEL(K)) I=K+1
      IF (I.LE.J) GO TO 200
C   DID WAVELENGTH MATCH A STANDARD IN AWAVEL(*) ?
      IF (I-1.GT.J) GO TO 600
C   WAVELENGTH NOT EXACTLY EQUAL TO ONE OF THE STANDARDS IN ARRAY
C   AWAVEL.  CHANGE WAVELENGTH TO EQUAL THE STANDARD IT IS CLOSEST TO.
C   THEN PRINT WARNING OF CHANGE.
      IF (WAVEL.GT.AWAVEL(K).AND.K.EQ.ILAST) GO TO 500
      IF (WAVEL.LT.AWAVEL(K).AND.K.EQ.IFIRST) GO TO 500
      IF (WAVEL-AWAVEL(K)) 300,600,400
C   WAVEL LT AWAVEL(K)
C   CHECK IF CLOSER TO AWAVEL(K) OR AWAVEL(K-1)
  300 DELTA1=WAVEL-AWAVEL(K-1)
      DELTA2=AWAVEL(K)-WAVEL
      IF (DELTA1.LT.DELTA2) K=K-1
      GO TO 500
C   WAVEL GT AWAVEL(K)
C   CHECK IF WAVEL CLOSER TO AWAVEL(K) OR AWAVEL(K+1)
  400 DELTA1=WAVEL-AWAVEL(K)
      DELTA2=AWAVEL(K+1)-WAVEL
      IF (DELTA2.LT.DELTA1) K=K+1
C   PRINT WARNING
  500 WAVEL=AWAVEL(K)
      WRITE (IOOUT,800) TWAVEL,WAVEL
C   LOAD LASER ID NUMBER
  600 LID=AID(K)
C   LOAD LINE NAME
      KK=2*K-1
      LNAME1=INAME1(KK)
      LNAME3=INAME1(KK+1)
      LNAME2=INAME2(K)
C   LOAD WAVELENGTH
      WAVEL=AWAVEL(K)
  700 RETURN
C
  800 FORMAT (29H *** WARNING INPUT WAVELENGTH,F7.3,
     1        11H CHANGED TO,F7.3,18H NEAREST STANDARD ,
     2        14HWAVELENGTH ***)
  900 FORMAT (24H  *** ERROR LASER LINE #,4(A2),6H# NOT ,
     1        9HVALID ***,/,26H CONTROL RETURNED TO MAIN ,
     2        12HFROM LZTRAN.)
      END
```

```
      SUBROUTINE DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)
C
C*******************************************************************
C
C
C     PURPOSE
C         DIRTRAN-2 EXPLOSION PRODUCED AND VEHICLE GENERATED DUST MODEL
C
C     INPUT,OUTPUT AND CALLING PROGRAM
C
C     INPUTS
C
C        VALUES IN ARGUMENT LIST
C
C     ICLMAT  INTEGER VALUE USED TO INDICATE HOW METEORLOGICAL DATA IS TO
C           BE MADE AVAILABLE  IF ICLMAT IS
C               0 - MET1 IDENTIFIER WITH THE APPROPRIATE PARAMETERS ARE TO
C                   BE READ IN
C               1 - NECESSARY METEORLOGICAL DATA IS PASSED IN COMMON/CLYMAT/
C                   AND MET1 IS NOT TO BE READ IN
C
C        WAVE1       WAVELENGTH IN MICROMETERS.  USED TO DETERMINE NWL.
C                    WHERE NWL IS AN INTEGER INDEX FOR WAVELENGTH DETERMINED
C                    WITHIN THE CODE
C                        NWL         WAVE1
C                         1   FOR  0.4 - 0.7 MICROMETER  (VISIBLE)
C                         2   FOR  0.8 - 1.1 MICROMETER
C                         3   FOR  3.5 - 4.0 MICROMETER
C                         4   FOR  8.5 - 12.0 MICROMETER
C                         5   FOR  2100 - 3200 MICROMETER
C
C     INPUTS TO BE READ
C
C        EACH INPUT RECORD BEGINS WITH A 4 LETTER IDENTIFIER IN COLUMNS
C     1-4 FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, UP TO 9, 8 COLUMNS
C     PER FIELD BEGINNING IN COLUMN 9.
C        THE INPUT FILE MAY CONTAIN SEVERAL SEQUENCES OF THE FOLLOWING
C     RECORDS.  EACH SEQUENCE SEPERATED BY A GO CARD.  ONCE THE INITIAL
C     SEQUENCE HAS BEEN READ IN AND THE MINIMUM REQUIREMENTS FOR EXECUTION
C     OF THE DESIRED OPTION HAS BEEN SATISFIED,ANY FOLLOWING SEQUENCE MAY
C     CONTAIN A SUBSET OF THE INITIAL RECORDS REDEFINING INPUT VARIABLES
C     AS DESIRED OR MAY CONTAIN A COMPLETELY NEW SET OF RECORDS.
C
C     ** EACH SET OF INPUTS MUST END WITH A DONE CARD**
C
C    RECORD 1
C     MET1
C
C        NATMOS      INTEGER WITH VALUES 1 TO 6 CORRESPONDING TO PASQUILL
C                    CATEGORIES A TO F.
C
C        ZTMP        THE HEIGHT AT WHICH A TEMPERATURE MEASUREMENT IS
C                    AVAILABLE.   VALID RANGE  0.5 - 100.0 M.
C
C        TMPMES      THE TEMPERATURE MEASURED IN DEGREES KELVIN TAKEN AT
C                    HEIGHT ZTMP.   VALID RANGE 270.0 - 315.0.
C
C        ZWND       THE HEIGHT AT WHICH A WIND SPEED MEASUREMENT IS
C                   AVAILABLE.  VALID RANGE  0.5 - 100.0 M.
C
C        UNDMES      THE WIND SPEED IN METERS/SECOND MEASURED AT ZWND
C                    VALID RANGE  .1 - 20.0 M/S
C
C        THWND       THE ANGLE THAT THE WIND VELOCITY VECTOR MAKES
C                    WITH THE USER'S POSITIVE X AXIS MEASURED IN DEGREES
C                    COUNTERCLOCKWISE. WHERE THE USERS POSITIVE X-AXIS
C                    POINTS EAST. THUS THWND IS THE ANGLE THAT THE WIND
C                    VELOCITY VECTOR MAKES WITH THE EAST.
C                    VALID RANGE:  -360.0 - 360.0 DEGREES.
C                    NOTE: THWND IS NOT NEEDED FOR OPTION 3
```

```
C
C RECORD 2
C  MET2
C
C     ID         A FLAG TO INDICATE WHETHER THE INVERSION LAYER HEIGHT
C                IS GROWING OR NOT.  IF ID IS
C                  0.  THE INVERSION LAYER HEIGHT IS RELATIVELY CONSTANT
C                  1.  THE INVERSION LAYER HEIGHT IS GROWING
C
C     PHI        THE LATITUDE OF THE DETONATION SITE.
C                VALID RANGE 1.0 - 90. DEGREES.  THAT IS THE NORTHERN
C                HEMISPHERE.
C
C RECORD 3
C  SOIL
C
C     NSOIL      INTEGER INDEX OF SOIL TYPE.  NSOIL IS
C                  1.  FOR SOIL-1, (DATA GRAF-II) EXPLOSIONS ONLY.
C                  2.  FOR SOIL-2, (DATA DIRT-I) EXPLOSIONS ONLY.
C                  3.  FOR SOIL-3, (DATA SMOKEWEEK-II) VEHICLES ONLY.
C
C     DSOD       DEPTH OF SOD IN METERS
C                VALID RANGE: 0.0 - 1.0 M.
C                NOTE: FOR VEHICLE MODEL IF DSOD>0.0 NO DUST IS
C                GENERATED
C
C     SILT       SILT CONTENT OF SOIL (PARTICLE DIAMETERS < 75 MICRONS)
C                I.E.   SILT=.15 INDICATES A SILT CONTENT OF 15%
C                NOTE: THIS INPUT NEEDED ONLY FOR VEHICLE MODEL (IOPT=5)
C
C RECORD 4
C  CHAR
C
C     NCHRG      CHARGE TYPE INDEX WITH FOLLOWING VALUES
C                     1.     SURFACE - LIVE FIRE OR 30 DEGREE TILTED
C                           STATIC, TIP ON GROUND
C                     2.     BARE CHARGE ON SURFACE
C                     3.     30 DEGREE TILTED TIP AT 0.3 METER DEPTH
C                     4.     30 DEGREE TILTED TIP AT 0.6 METER DEPTH
C                     5.     HORIZONTAL PROJECTILE ON SURFACE
C                 DEFAULT VALUE IS 1 IF NCHRG IS NOT BETWEEN 1 AND 5.
C
C     CHWT       THE WEIGHT OF THE CHARGE IN KG-TNT.
C                VALID RANGE:  0.1 - 100.0 KG-TNT.
C
C     DETDEP     THE DEPTH OF DETONATION IN METERS.
C                VALID RANGE:  0.0 - 2.0 M.
C
C RECORD 5
C  EXPL
C
C     NARY       TYPE OF CHARGE DISTRIBUTION (USED FOR PROPER INPUT AND
C                OUTPUT FORMATS) IF THE VALUE OF NARY AND IOPT OF THE GO
C                CARD ARE NOT COMPATIBLE CATASTROPHE COULD RESULT!
C                NOTE: WHEN NARY IS
C                        1.    IOPT MUST ALSO BE 1.
C                        2.    IOPT MUST ALSO BE 2.
C                        3.    IOPT MUST BE      4.
C              1.-SIMULTANEOUS BURST,UNIFORMLY DISTRIBUTED CHARGES
C                IN A PARALLELOGRAM.
C                (SPECIAL CASES ARE ,SINGLE CHARGE ,RECTANGLE AND
C                 ZIG ZAG PATTERN)
C
C              2.-SIMULTANEOUS BURST,RANDOMLY DISTRIBUTED CHARGES.
C
C              3.-SEQUENTIAL IN TIME AND RANDOM IN SPACE DISTRIBUTION OF
C                CHARGES.
C
C                NOTE:
C                  WHEN NARY=2. EACH CHARGE LOCATION MUST BE SPECIFIED
```

```
C                     WHEN NARY=3. EACH CHARGE LOCATION AND DETONATION TIME
C                     MUST BE SPECIFIED
C                     THE CHARGE LOCATIONS ARE INPUT DIRECTLY FOLLOWING
C                     THIS INPUT RECORD WITH ONE LOCATION (AND DETONATION
C                     TIME IF APPROPRIATE) PER RECORD.
C
C     NCHS          SINGLY DIMENSIONED ARRAY SPECIFING NUMBER OF CHARGES
C                   WITH THE MAXIMUM TOTAL OF CHARGES 200.
C                   WHEN NARY=1. NCHS(1) IS THE NUMBER OF CHARGES IN THE
C                   DIRECTION OF SIDE1 AND NCHS(2) IS THE NUMBER OF
C                   CHARGES IN THE DIRECTION OF SIDE2. FOR A SINGLE CHARGE
C                   SET NCHS(1)=NCHS(2)=1..
C                   WHEN NARY=2. OR 3. NCHS(1) IS THE TOTAL NUMBER OF
C                   CHARGES AND SET NCHS(2)=1..
C
C     SRCBAS        SINGLY DIMENSIONED ARRAY CONTAINING THE COORDINATES OF
C                   A CORNER POINT OF THE BOUNDING PARALLELOGAM WHEN
C                   NARY=1.  AND IS ALSO USED AS THE REFERENCE CHARGE
C                   BY THE OBSERVER. THAT IS, SRCBAS IS THE ORIGIN OF THE
C                   OBSERVERS COORDINATE SYSTEM.  WHEN NARY=2. OR 3.
C                   SRCBAS SHOULD NOT APPEAR ON THE INPUT FILE AS COOR(I,1)
C                   IS USED AS THE REFERENCE CHARGE. WHERE COOR(I,1) IS
C                   THE FIRST CHARGE LOCATION SPECIFIED.
C                   VALID RANGE -10000.0 - 10000.0
C
C     SIDE1         SINGLY DIMENSIONED ARRAY NECESSARY ONLY WHEN NARY=1.,
C                   SPECIFING ONE SIDE OF THE BOUNDING PARALLELOGRAM
C                   FROM THE POINT SRCBAS(I). THAT IS, SIDE1 IS A VECTOR
C                   TO THE NEXT CHARGE ALONG ONE SIDE OF THE PARALLELOGRAM.
C                   WHEN NARY=2. OR 3. THIS VARIABLE SHOULD NOT APPEAR ON
C                   THE INPUT FILE.
C
C     SIDE2         SINGLY DIMENSIONED ARRAY NECESSARY ONLY WHEN NARY=1.
C                   SPECIFING A SECOND SIDE OF THE BOUNDING PARALLELOGRAM
C                   FROM THE POINT SRCBAS(I).THAT IS SIDE2 IS A VECTOR TO
C                   TO THE NEXT CHARGE ALONG THE SECOND SIDE OF THE
C                   PARALLELOGRAM.
C                   WHEN NARY=2. OR 3. THIS VARIABLE SHOULD NOT APPEAR ON
C                   THE INPUT FILE.
C
C   ** THE FOLLOWING RECORD MUST APPEAR THE APPROPRIATE NUMBER OF TIMES
C   ** IMEDIATELY FOLLOWING THE ABOVE RECORD IF NARY IS 2. OR 3. ON THE
C   ** ABOVE RECORD.  THAT IS IT MUST APPEAR THE SAME NUMBER OF TIMES
C   ** AS THER ARE CHARGES AS SPECIFIED ON RECORD EXPL. (I.E. IF NARY=2.
C   ** AND NCHS=5. THE THIS RECORD MUST APPEAR 5 TIMES.
C
C RECORD 6
C  LOCA
C
C     COOR          DOUBLY DIMENSIONED ARRAY CONTAINING THE DETONATION
C                   COORDINATES FOR EACH CHARGE WHEN NARY=2. OR 3.. WHEN
C                   NARY=1. THIS VARIABLE NEED NOT BE SPECIFIED AS THE
C                   CHARGE LOCATIONS ARE CALCULATED IN THE CODE FROM
C                   NCHS,SIDE1,SIDE2.
C
C     TSTAG         SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF
C                   DETONATION OF EACH CHARGE. THIS IS ONLY SPECIFIED WHEN
C                   NARY=3..
C
C RECORD 7
C  VEHC
C
C     V0            DOUBLY DIMENSIONED ARRAY CONTAINING THE INITIAL
C                   POSITION OF THE VEHICLE. V0(1)=X-COORDINATE
C                   V0(2)=Y-COORDINATE.
C                   VALID RANGE: -10000.0 - 10000.0
C
C     VEHDIR        VEHICLE DIRECTION  THE ANGLE THAT THE VEHICLE VELOCITY
C                   VECTOR MAKES WITH THE USER'S POSITIVE X-AXIS MEASURED
C                   IN DEGREES COUNTERCLOCKWISE. WHERE THE USER'S POSITIVE
```

```
C                    X-AXIS POINTS EAST. THUS VEHDIR IS THE ANGLE THE
C                    VELOCITY VECTOR MAKES WITH THE EAST.
C                    VALID RANGE:  -360.0 - 360.0
C
C     VEHSPD         VEHICLE SPEED IN  M/S
C
C     VEHWID         VEHICLE WIDTH IN METERS
C
C     VEHWHT         VEHICLE WEIGHT IN KGS.
C
C     VEHTYP         TRACTION MECHANISM
C
C                   =0.  VEHICLE HAS TIRES
C                   =1.  VEHICLE IS TRACKED
C
C RECORD 8
C  TRNC
C
C     TRNCOR         A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE
C                    COORDINATES OF THE TRANSMITTER.  THE COORDINATE SYSTEM
C                    MUST BE IN METERS.  THE THIRD COORDINATE IS RESTRICTED
C                    TO BE BETWEEN .5 AND 10000.0 METERS (HEIGHT).
C                    VALID RANGE OF THE FIRST TWO COORDINATES:
C                    -10000.0  -  10000.0 M.
C
C  ** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE
C  ** ARRAY TRNCOR NEED NOT BE SPECIFIED.
C
C     TRNMIN         VALUE SUCH THAT A TRANSMITTANCE BELOW THIS VALUE CAN
C                    BE CONSIDERED ZERO.  DEFAULT IS 1.E-05
C                    VALID RANGE: 1.0 - 1.E-05
C
C RECORD 9
C  RECC
C
C     RECCOR         A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE
C                    COORDINATES OF THE RECEIVER.  (METERS)
C                    THE THIRD COORDINATE IS RESTRICTED TO BE BETWEEN
C                    .5 AND 10000.0 METERS.  VALID RANGE OF THE FIRST TWO
C                    COORDINATES IS:  -10000.0  -  10000.0 M.
C
C  ** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE
C  ** ARRAY RECCOR NEED NOT BE  SPECIFIED.
C
C RECORD 10
C  OBSC
C
C     OBSCOR         A SINGLY DIMENSIONED ARRAY CONTAINING THE X AND Y
C                    COORDINATES, RESP., OF THE OBSERVER. (METERS)
C                    VALID RANGE:   -10000.0  -  10000.0
C
C     SPCHT          A SPECIFIED HEIGHT IN METERS AT WHICH THE WIDTH OF
C                    THE CLOUD AS VIEWED FROM POSITION OBSCOR IS DESIRED.
C                    MUST BE BETWEEN 1. AND 5. METERS.
C
C  ** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE
C  ** ARRAY OBSCOR AND VARIABLE SPCHT NEED NOT BE SPECIFIED.
C
C RECORD 11
C  TIMS
C
C     TSTART         TIME AFTER DETONATION TO START TRANSMITTANCE AND/OR
C                    CLOUD DIMENSION CALCULAIONS
C                    VALID RANGE: .5 - 1000.0 SEC.
C
C     TEND           TIME AFTER DETONATION TO TERMINATE TRANSMITTANCE
C                    AND/OR CLOUD DIMENSIONS.
C                    VALID RANGE: .5-1000.0 SEC. (TEND MUST BE .GE. TSTART)
C
C     TINC           TIME INCREMENT BETWEEN CALCULATIONS
```

```
C
C RECORD 12
C  GO      ** THIS CARD INDICATES THAT THIS SEQUENCE OF INPUTS ARE
C             COMPLETE AND CALCULATIONS ARE TO BEGIN.
C
C      IOPT   OPTION TO BE USED
C
C             1. SIMULTANEOUS BURST ,UNIFORMLY DISTRIBUTED CHARGES IN A
C                PARALLELOGRAM
C
C             2. SIMUTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES
C
C             3. THE CODE IS TO PRECOMPUTE A SINGLE CLOUD AND STORE ON
C                AN EXTERNAL FILE FOR USE LATER
C
C             4. THE CODE IS TO USE A CLOUD THAT HAS BEEN PRECOMPUTED
C                (NO CLOUD DIMENSIONS ARE COMPUTED FOR THIS OPTION)
C
C             5. VEHICLE DUST MODEL
C
C      IFILE  FORTRAN LOGICAL UNIT TO WHICH THE CODE IS TO WRITE FOR
C             OPTION 3 OTHERWISE IT NEED NOT BE SPECIFIED
C
C RECORD 13
C  DONE    ** THIS RECORD INDICATES THAT THE USER HAS COMPLETED HIS
C            DESIRED SEQUENCE OF INPUTS AND ALL CALCULATIONS ARE
C            TERMINATED
C
C
C  OUTPUTS
C
C      ZINV       THE ESTIMATED INVERSION HEIGHT.
C
C      TRNLOS     THE TRANSMITTANCE ALONG THE LINE OF SIGHT BETWEEN
C                 THE TRANSMITTER AND THE RECEIVER.
C
C      IERR       INTEGER ERROR CODE WHICH EQUALS 1 IF A FATAL ERROR
C                 OCCURS AND 0 OTHERWISE
C
C      NERR        INTEGER ERROR CODE WITH THE VALUES
C                    0     NO ERRORS
C                    4     NO TRANSMITTER AND RECEIVER OR OBSERVER
C                          COORDINATES WERE SPECIFIED SO NO RESULTS WERE
C                          CALCULATED.
C                    7     THE CALCULATION OF ATMOSPHERIC PARAMETERS DID
C                          NOT CONVERGE.
C
C
C      CNTRD      A SINGLY DIMENSIONED ARRAY CONTAINING THE HORIZONTAL
C                 COORDINATE AND THE VERTICAL COORDINATE OF THE
C                 CENTROID OF THE CLOUD.
C
C      HEIGHT     THE HEIGHT OF THE CLOUD IN METERS.
C
C      CENWTH     THE WIDTH OF THE CLOUD IN METERS AT THE CENTROID
C                 HEIGHT
C
C      SPCWTH     THE WIDTH OF THE CLOUD IN METERS AT THE SPECIFIED
C                 HEIGHT
C
C      NCPTS      THE NUMBER OF POINTS DETERMINED ON THE EDGE OF THE
C                 CLOUD.
C
C      CPTS       A DOUBLY DIMENSIONED ARRAY CONTAINING THE COORDINATES
C                 OF POINTS ON THE EDGE OF THE CLOUD.  CPTS(1,J)
C                 IS THE HORIZONTAL COORDINATE OF THE J-TH POINT
C                 AND CPTS(2,J)IS THE VERTICAL COORDINATE OF THE
C                 J-TH POINT.  THE FIRST INDEX MUST BE DIMENSIONED
C                 TO 2.
C
```

```
C       SUBROUTINES CALLED
C
C       DUSTCL   CONTROLING ROUTINE FOR THE CALCULATION OF CLOUD DIMENSIONS
C                AND TRANSMITTANCES THROUGH DUST CLOUDS FOR OPTIONS 1 AND 2
C                GIVEN METEORLOGICAL DATA, SOIL AND EXPLOSIVE
C                CHARACTERISTICS,AND WAVELENGTH.
C
C       COMPCL   CONTROLING ROUTINE  FOR PRECOMPUTING A SINGLE EXPLOSION
C                (OPTION 3) GIVEN METEOROLOGICAL DATA, SOIL AND EXPLOSION
C                CHARACTERISTICS.  ALSO USES THIS PRECOMPUTED CLOUD AT SOME
C                LATER RUNNING OF THE CODE (OPTION 4) TO ESTIMATE A
C                TRANSMITTANCE GIVEN TRANSMITTER AND RECEIVER COODINATES.
C
C       VEHCL    CONTROLING ROUTINE FOR THE CALCULATION OF A TRANSMITTANCE
C                THROUGH A VEHICLE GENERATED DUST CLO"D (OPTION 5) GIVEN
C                METEORLOGICAL DATA, SOIL CHARACTERISTICS, VEHICLE
C                CHARACTERISTICS, AND WAVELENGTH.
C
C*****************************************************************
      LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM,      CLMRED,DHDT,ONCE
      LOGICAL TEST,NEWVEH,NEWCOR
      LOGICAL M1,M2,SL,CH,EX,TC,RC,OC,TM,VH
      INTEGER VEHTYP
      REAL M,N
      DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2),TRNCOR(3)
      DIMENSION SRCBAS(2),SIDE1(2),SIDE2(2),NCHS(2)
     1   ,RECCOR(3),CPTS(2,6),CNTRD(2),OBSCOR(2)
      DIMENSION RDIN(10),RKEY(12),V0(2),PAS(6)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /CLYMAT/ TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
     1 FOGPRB,WNDVEL,WINDIR,IPASCT
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +              COOR(2,200),TSTAG(200),DMMY(401)
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON /GEOMET/PTS(15),IGEOSW
      COMMON/OPTION/IOPT,IFILE
      DATA RKEY/4HMET1,4HMET2,4HSOIL,4HCHAR,4HEXPL,4HVEHC,4HTRNC,
     1          4HRECC,4HOBSC,4HTIMS,4HGO  ,4HDONE/
      DATA PAS/4HA    ,4HB   ,4HC   ,4HD   ,4HE   ,4HF    /
      DATA M1,M2,SL,CH,EX,TC,RC,OC,TM,VH/.FALSE.,.FALSE.,.FALSE.,
     1.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE./
      DATA NEWATM,NEWSRC,NEWVEH,LOSTRN,EDGE,NEWTIM/.FALSE.,.FALSE.,
     1.FALSE.,.FALSE.,.FALSE.,.FALSE./
      DATA NEWCOR/.FALSE./
      DATA VISEXT,RTPI/.1,1.772454/
      IERR=0
      CLMRED=.FALSE.
      DHDT=.FALSE.
      ONCE=.FALSE.
      TEST=.FALSE.
      WRITE(IOOUT,800)
  800 FORMAT(1H0,36X,42HDIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION,
     1 15H    CALCULATION,//,36X,60H*** NOTE -- ALL UNITS ARE MKS UNLESS
     2 OTHERWISE SPECIFIED ***,//)
      DO 5 K=1,200
      TSTAG(K)=0.0
    5 CONTINUE
C
C  DETERMINE INTEGER INDEX FOR WAVELENGTH
C
   10 IF(WAVE1.LT.0.4)GO TO 29
      IF(WAVE1.GT.0.7)GO TO 21
      NWL=1
      GO TO 30
   21 IF(WAVE1.LT.0.8)GO TO 29
      IF(WAVE1.GT.1.1)GO TO 22
      NWL=2
      GO TO 30
```

```
   22 IF(WAVE1.LT.3.5)GO TO 29
      IF(WAVE1.GT.4.0)GO TO 23
      NWL=3
      GO TO 30
   23 IF(WAVE1.LT.8.5)GO TO 29
      IF(WAVE1.GT.12.0)GO TO 24
      NWL=4
      GO TO 30
   24 IF(WAVE1.LT.2100.)GO TO 29
      IF(WAVE1.GT.3200.)GO TO 29
      NWL=5
      GO TO 30
   29 WRITE(IOOUT,802)
  802 FORMAT(37X,38H*** DIRTRAN ERROR - WAVE1 OUT OF RANGE)
      IERR=1
      GO TO 999
   30 CONTINUE
C
C  READ DATA AND STORE APPROPRIATELY
C
      DO 300 II=1,15
      IF(II.EQ.15)GO TO 900
      READ(IDIN,700)(RDIN(J),J=1,10)
  700 FORMAT(A4,4X,9F8.2)
      IF(RDIN(1).EQ.RKEY(1))GO TO 50
      IF(RDIN(1).EQ.RKEY(2))GO TO 70
      IF(RDIN(1).EQ.RKEY(3))GO TO 90
      IF(RDIN(1).EQ.RKEY(4))GO TO 110
      IF(RDIN(1).EQ.RKEY(5))GO TO 130
      IF(RDIN(1).EQ.RKEY(6))GO TO 150
      IF(RDIN(1).EQ.RKEY(7))GO TO 170
      IF(RDIN(1).EQ.RKEY(8))GO TO 190
      IF(RDIN(1).EQ.RKEY(9))GO TO 210
      IF(RDIN(1).EQ.RKEY(10))GO TO 230
      IF(RDIN(1).EQ.RKEY(11))GO TO 310
      IF(RDIN(1).EQ.RKEY(12))GO TO 999
      WRITE(IOOUT,804)
  804 FORMAT(33X,52H***DIRTRAN-2 ERROR, INPUT DOES NOT CONFORM TO PROPER
     1       14H CONVENTION***)
      WRITE(IOOUT,806)(RDIN(J),J=1,9)
  806 FORMAT(26X,A4,4X,9F8.2)
      GO TO 999
C
C  STORE AND PRINT OUT ATMOSPHERIC CONDITIONS
C
   50 CONTINUE
      M1=.TRUE.
      NIO=1
      IF(ICLMAT.EQ.1)GO TO 55
      NATMOS=IFIX(RDIN(2))
      ZTMP(1)=RDIN(3)
      TMPMES(1)=RDIN(4)
      ZWND(1)=RDIN(5)
      WNDMES(1)=RDIN(6)
      THWND=RDIN(7)
      GO TO 60
C
C  IPSCAT   PASQUILL CATEGORY
C  WNDVEL   WIND VELOCITY IN M/S MEASURED AT 2 M. ABOVE GROUND
C  WINDIR   WIND DIRECTION IN DEGREES   CLOCKWISE FROM TRUE NORTH
C  TEMP     TEMPERATURE IN DEGREES C MEASURED AT 2 M. ABOVE GROUND
C
   55 NATMOS=IPASCT
      ZTMP(1)=2.
      ZWND(1)=2.
      WNDMES(1)=WNDVEL
      TMPMES(1)=TEMP+273.0
      THWND=270.0-WINDIR
   60 CONTINUE
      WRITE(IOOUT,808)
```

```
 808  FORMAT(1X)
      WRITE(IOOUT,810)PAS(NATMOS)
  810 FORMAT(50X,28HPASQUILL CATEGORY           ,A4)
      WRITE(IOOUT,812)(ZTMP(I),TMPMES(I),ZWND(I),WNDMES(I),I=1,NIO)
  812 FORMAT(36X,4H HT ,F8.2,7H  TEMP ,F8.2,7H     HT,F8.2,7H  WIND ,
     1 F8.2)
      WRITE(IOOUT,814)THWND
  814 FORMAT(51X,22H  WIND DIRECTION      ,F8.2)
      GO TO 300
   70 CONTINUE
      M2=.TRUE.
      ID=IFIX(RDIN(2))
      IF(ID.NE.0)DHDT=.TRUE.
      PHI=RDIN(3)
      WRITE(IOOUT,819)PHI
      IF(DHDT)WRITE(IOOUT,816)
  816 FORMAT(47X,37HTHE INVERSION LAYER HEIGHT IS GROWING)
      IF(.NOT.DHDT)WRITE(IOOUT,818)
  818 FORMAT(47X,38HTHE INVERSION LAYER HEIGHT IS CONSTANT)
  819 FORMAT(/,52X,20HLATITUDE            ,F8.2)
      GO TO 300
C
C  STORE AND WRITE SOIL CHARACTERISTICS
C
   90 CONTINUE
      SL=.TRUE.
      NSOIL=IFIX(RDIN(2))
      DSOD=RDIN(3)
      SILT=RDIN(4)
      IF(NSOIL.EQ.1)WRITE(IOOUT,821)
  820 FORMAT(56X,15HSILT CONTENT   ,F5.2)
  821 FORMAT(/,63X,6HSOIL-1)
      IF(NSOIL.EQ.2)WRITE(IOOUT,822)
  822 FORMAT(/,63X,6HSOIL-2)
      IF (NSOIL.EQ.3) WRITE (IOOUT,710)
 710  FORMAT(/,23X,8H  SOIL-3)
      IF(SILT.GT.1.E-06)WRITE(IOOUT,820)SILT
      WRITE(IOOUT,823)DSOD
  823 FORMAT(53X,21H  DEPTH OF SOD       ,F5.2)
      IF (NSOIL.LT.1.OR.NSOIL.GT.2) NSOIL=2
      GO TO 300
C
C  STORE AND WRITE EXPLOSIVE CHARGE CHARACTERISTICS
C
  110 CONTINUE
      CH=.TRUE.
      NCHRG=IFIX(RDIN(2))
      IF(NCHRG.LT.1.OR.NCHRG.GT.5)NCHRG=1
      CHWT=RDIN(3)
      DETDEP=RDIN(4)
      IF(NCHRG.EQ.1)WRITE(IOOUT,824)
  824 FORMAT(/,35X,47HSURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC,,
     1       14H TIP ON GROUND)
      IF(NCHRG.EQ 2)WRITE(IOOUT,825)
  825 FORMAT(/,55X,22HBARE CHARGE ON SURFACE)
      IF(NCHRG.EQ.3)WRITE(IOOUT,826)
  826 FORMAT(/,46X,39H30 DEGREE TILTED TIP AT 0.3 METER DEPTH)
      IF(NCHRG.EQ.4)WRITE(IOOUT,827)
  827 FORMAT(/,46X,39H30 DEGREE TILTED TIP AT 0.6 METER DEPTH)
      IF(NCHRG.EQ.5)WRITE(IOOUT,828)
  828 FORMAT(/,50X,32HHORIZONTAL PROJECTILE ON SURFACE)
      WRITE(IOOUT,829)CHWT
  829 FORMAT(45X,30HWEIGHT OF CHARGE             ,F8.2,4H KG.)
      WRITE(IOOUT,830)DETDEP
  830 FORMAT(47X,30HDETONATION DEPTH             ,F8.2)
      GO TO 300
C
C  STORE AND WRITE OUT INFORMATION ABOUT THE DETONATION LOCATIONS
C
  130 CONTINUE
```

```
      EX=.TRUE.
      NARY=IFIX(RDIN(2))
      NCHS(1)=IFIX(RDIN(3))
      NCHS(2)=IFIX(RDIN(4))
      SRCBAS(1)=RDIN(5)
      SRCBAS(2)=RDIN(6)
      SIDE1(1)=RDIN(7)
      SIDE1(2)=RDIN(8)
      SIDE2(1)=RDIN(9)
      SIDE2(2)=RDIN(10)
      IF(NARY.EQ.2)GO TO 133
      IF(NARY.EQ.3)GO TO 136
C
C  CHARGE DISTRIBUTION TYPE 1
C
      WRITE(IOOUT,831)
  831 FORMAT(/,31X,42HSIMULTANEOUS BURST , UNIFORMLY DISTRIBUTED,
     +27H CHARGES IN A PARALLELOGRAM)
      NCH=NCHS(1)*NCHS(2)
      WRITE(IOOUT,832)NCH,(SRCBAS(I),I=1,2)
  832 FORMAT(27X,28HTOTAL NUMBER OF CHARGES IS ,1X,I3,1X,
     +27H WITH REFERENCE CHARGE AT (,F8.2,1H,,F8.2,1H))
      WRITE(IOOUT,834)NCHS(1),(SIDE1(I),I=1,2)
  834 FORMAT(32X,I3,1X,45HCHARGES WITH DIRECTION AND SPACING GIVEN BY (,
     +F8.2,1H,,F8.2,1H))
      WRITE(IOOUT,834)NCHS(2),(SIDE2(I),I=1,2)
      GO TO 300
C
C  CHARGE DISTRIBUTION TYPE 2
C
  133 NCH=NCHS(1)
      DO 134 J=1,NCH
      READ(IOIN,701)(COOR(K,J),K=1,2)
  701 FORMAT(8X,2F8.2)
  134 CONTINUE
      WRITE(IOOUT,836)
  836 FORMAT(/,42X,48HSIMULTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES)
      WRITE(IOOUT,838)NCHS(1)
  838 FORMAT(51X,26HTOTAL NUMBER OF CHARGES IS  ,1X,I3 )
      WRITE(IOOUT,840)
  840 FORMAT(55X,22HDETONATION COORDINATES)
      DO 135 J=1,NCH
      WRITE(IOOUT,842)(COOR(I,J),I=1,2)
  842 FORMAT(53X,2(3X,F8.2))
  135 CONTINUE
      GO TO 300
C
C  CHARGE DISTRIBUTION TYPE 3
C
  136 NCH=NCHS(1)
      DO 137 J=1,NCH
      READ(IOIN,702)(COOR(K,J),K=1,2),TSTAG(J)
  702 FORMAT(8X,3F8.2)
  137 CONTINUE
      WRITE(IOOUT,844)
  844 FORMAT(/,30X,38HSEQUENTIAL IN TIME AND RANDOM IN SPACE,
     +24H DISTRIBUTION OF CHARGES)
      WRITE(IOOUT,838)NCH
      WRITE(IOOUT,846)
  846 FORMAT(45X,25H DETONATION COORDINATES  ,7X,10HBLAST TIME)
      DO 138 J=1,NCH
      WRITE(IOOUT,848)(COOR(I,J),I=1,2),TSTAG(J)
  848 FORMAT(46X,F8.2,3X,F8.2,12X,F8.2)
  138 CONTINUE
      GO TO 300
C
C  STORE AND PRINT OUT INFORMATION ABOUT VEHICLE
C
  150 CONTINUE
      VH=.TRUE.
```

```
      V0(1)=RDIN(2)
      V0(2)=RDIN(3)
      VEHDIR=RDIN(4)
      VEHSPD=RDIN(5)
      VEHWID=RDIN(6)
      VEHWHT=RDIN(7)
      VEHTYP=IFIX(RDIN(8))
      WRITE(IOOUT,850)V0(1),V0(2)
  850 FORMAT(//,44X,26HINITIAL VEHICLE POSITION (,F8.2,1H,,F8.2,1H))
      WRITE(IOOUT,852)VEHDIR
  852 FORMAT(44X,19HVEHICLE DIRECTION  ,F8.2,17H  (CCW FROM EAST))
      WRITE(IOOUT,854)VEHSPD
  854 FORMAT(50X,20HVEHICLE SPEED       ,F8.2,4H M/S)
      WRITE(IOOUT,856)VEHWID
  856 FORMAT(52X,20HVEHICLE WIDTH       ,F8.2)
      WRITE(IOOUT,858)VEHWHT
  858 FORMAT(52X,20HVEHICLE WEIGHT      ,F8.2)
      IF(VEHTYP.EQ.0)WRITE(IOOUT,891)
  891 FORMAT(58X,15HWHEELED VEHICLE)
      IF(VEHTYP.EQ.1)WRITE(IOOUT,892)
  892 FORMAT(58X,15HTRACKED VEHICLE)
      GO TO 300
C
C  STORE TRANSMITTER COORDINATES AND TRANSMISSION TRHESHOLD
C
  170 CONTINUE
      TC=.TRUE.
      NEWCOR=.TRUE.
      TRNCOR(1)=RDIN(2)
      TRNCOR(2)=RDIN(3)
      TRNCOR(3)=RDIN(4)
      TRNMIN=RDIN(5)
      IF(TRNMIN.LT.1.E-05)TRNMIN=1.E-05
      THRESH=-ALOG(TRNMIN)
      GO TO 300
C
C  STORE RECEIVER COORDINATES
C
  190 CONTINUE
      RC=.TRUE.
      RECCOR(1)=RDIN(2)
      RECCOR(2)=RDIN(3)
      RECCOR(3)=RDIN(4)
      GO TO 300
C
C  STORE OBSERVER COORDINATES
C
  210 CONTINUE
      OC=.TRUE.
      OBSCOR(1)=RDIN(2)
      OBSCOR(2)=RDIN(3)
      SPCHT=RDIN(4)
      GO TO 300
C
C  STORE TIME INTERVAL FOR CALCULATIONS
C
  230 CONTINUE
      TM=.TRUE.
      TSTART=RDIN(2)
      TEND=RDIN(3)
      TINC=RDIN(4)
      IF (TINC.LE.0.0) TINC=1.
      IF(TEND.LT.TSTART)GO TO 903
      LIM=IFIX((TEND-TSTART)/TINC)+1
  300 CONTINUE
 310  CONTINUE
      IF(IGEOSW.NE.1) GO TO 333
      TRNCOR(1)=PTS(1)*1000.
      TRNCOR(2)=PTS(2)*1000.
      TRNCOR(3)=PTS(3)*1000.
```

```
      RECCOR(1)=PTS(4)*1000.
      RECCOR(2)=PTS(5)*1000.
      RECCOR(3)=PTS(6)*1000.
      OBSCOR(1)=PTS(10)*1000.
      OBSCOR(2)=PTS(11)*1000.
      SPCHT=PTS(12)*1000.
  333 CONTINUE
      IOPT=IFIX(RDIN(2))
      IFILE=IFIX(RDIN(3))
      IF(IOPT.NE.3) GO TO 305
      WRITE(IOOUT,301)
      WRITE(IOOUT,302)
      WRITE(IOOUT,303) IFILE
      WRITE(IOOUT,304)
      WRITE(IOOUT,301)
  301 FORMAT(1H0,130(1H*),/)
  302 FORMAT(1H0,58X,15HDIRTRAN WARNING,/)
  303 FORMAT(1H0,38X,13HLOGICAL UNIT ,I2,27H IS ASSIGNED TO A TEMPORARY,
     + 13H STORAGE FILE)
  304 FORMAT(1H0,28X,46HCARE MUST BE TAKEN TO INSURE THAT THIS UNIT IS,
     +27H NOT IN USE BY ANOTHER FILE,/)
  305 CONTINUE
      IF((IOPT.EQ.1.OR.IOPT.EQ.2).AND.(EX.AND.(.NOT.CH)))GO TO 909
      IF((IOPT.EQ.1.OR.IOPT.EQ.2).AND.(CH.AND.(.NOT.EX)))GO TO 909
      IF((IOPT.EQ.3).AND.OC)OC=.FALSE.
      IF((IOPT.EQ.4).AND.OC)OC=.FALSE.
      IF((IOPT.EQ.5).AND.OC)OC=.FALSE.
      IF((TC.AND.(.NOT.RC)).OR.(RC.AND.(.NOT.TC)))GO TO 913
      IF(M1. OR.M2)NEWATM=.TRUE.
      IF(IOPT.EQ.3.AND.CH)NEWSRC=.TRUE.
      IF(EX.AND.CH)NEWSRC=.TRUE.
      IF(TC.AND.RC)LOSTRN=.TRUE.
      IF(OC)EDGE=.TRUE.
      IF(VH)NEWVEH=.TRUE.
      IF(IOPT.EQ.3)LIM=1
      DO 400 J=1,LIM
      TIME=TSTART+TINC*FLOAT(J-1)
      NEWTIM=.TRUE.
      NERR=0
      IF(IOPT.EQ.1.OR.IOPT.EQ.2)GO TO 320
      IF(IOPT.EQ.3.OR.IOPT.EQ.4)GO TO 325
      IF((IOPT.EQ.1).AND.(NARY.NE.1))GO TO 915
      IF((IOPT.EQ.2).AND.(NARY.NE.2))GO TO 915
      IF((IOPT.EQ.4).AND.(NARY.NE.3))GO TO 915
C
C  COMPUTE FOR VEHICLE SOURCE
C
C  CHECK TO SEE IF WE HAVE THE MINIMUM INPUT REQUIREMENTS
C
      IF(DSOD.GT.0.0)GO TO 315
      IF(.NOT.(M1.AND.M2.AND.SL.AND.VH.AND.TC.AND.RC.AND.TM))GO TO 911
      CALL VEHCL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,PHI,NSOIL,
     1                SILT,NWL,TRNCOR,RECCOR,TIME,DHDT,V0,VEHDIR,
     2           VEHSPD,VEHWID,VEHWHT,VEHTYP,NEWATM,NEWVEH,TRNLOS,NERR)
      NEWVEH=.FALSE.
      NEWATM=.FALSE.
      GO TO 330
  315 TRNLOS=1.0
      GO TO 335
  325 CONTINUE
      IF(IOPT.EQ.4.AND.(.NOT.(EX.AND.TC.AND.RC.AND.TM)))GO TO 911
      IF(IOPT.EQ.3.AND.(.NOT.(M1.AND.M2.AND.SL.AND.CH)))GO TO 911
      CALL COMPCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,
     1            PHI,NEWSRC,CHWT,NCHRG,NCHS,DETDEP,NSOIL,DSOD,NWL,
     2            TRNCOR,RECCOR,TIME,DHDT,TRNLOS,NERR)
      NEWSRC=.FALSE.
      NEWATM=.FALSE.
      IF(IOPT.EQ.3)GO TO 410
      GO TO 330
  320 CONTINUE
```

```
C
C    CHECK TO SEE IF MINIMUM INPUTS ARE AVAILABLE
C
      IF(.NOT.(M1.AND.M2.AND.SL.AND.CH.AND.EX.AND.(TC.AND.RC.OR.OC).AND.
     1TM))GO TO 911
      CALL DUSTCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,
     1            THWND,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,
     2            LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,
     3            TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS,
     4            NERR,NCHS,SRCBAS,SIDE1,SIDE2,DHDT)
      NEWSRC=.FALSE.
      NEWATM=.FALSE.
  330 IF(NERR.EQ.0)GO TO 335
      WRITE(IOOUT,857)NERR
  857 FORMAT(55X,30H ***** DIRTRAN ERROR NUMBER     ,I2)
      GO TO 400
  335 IF(ONCE.AND.(.NOT.NEWCOR))GO TO 340
      NEWCOR=.FALSE.
      IZINV=IFIX(ZINV)
      WRITE(IOOUT,859)IZINV
  859 FORMAT(//,47X,30HESTIMATED INVERSION HEIGHT     ,I7)
C
C   IF OBSERVER IS SPECIFIED, OUTPUT IS LABELED FOR EACH TIME.
C   IF ONLY TRANSMITTER AND RECEIVER ARE INPUT, OUTPUT IS TABULAR
C
      IF(TC.AND.RC.AND.OC)GO TO 350
      IF(OC)GO TO 350
      IF(.NOT.(TC.AND.RC))GO TO 905
      WRITE(IOOUT,860)WAVE1
  860 FORMAT(/,47X,18HWAVELENGTH        ,F7.2,12H MICROMETERS)
      WRITE(IOOUT,862)(TRNCOR(I),I=1,3)
      WRITE(IOOUT,864)(RECCOR(I),I=1,3)
  862 FORMAT(37X,28HTRANSMITTER COORDINATES      ,3F10.2)
  864 FORMAT(37X,28HRECEIVER    COORDINATES      ,3F10.2)
      WRITE(IOOUT,866)
  866 FORMAT(/52X,4HTIME,10X,13HTRANSMITTANCE)
  340 CONTINUE
      WRITE(IOOUT,868)TIME,TRNLOS
  868 FORMAT(52X,F8.2,10X,E10.5)
      ONCE=.TRUE.
      GO TO 400
  350 WRITE(IOOUT,923)TIME
  923 FORMAT(//,48X,28HTIME AFTER BLAST            ,F7.2)
      IF(.NOT.(TC.AND.RC))GO TO 360
      WRITE(IOOUT,808)
      WRITE(IOOUT,860)WAVE1
      WRITE(IOOUT,862)(TRNCOR(I),I=1,3)
      WRITE(IOOUT,864)(RECCOR(I),I=1,3)
      WRITE(IOOUT,870)TRNLOS
 870  FORMAT(42X,38HTRANSMITTANCE ALONG THE LINE OF SIGHT  ,E10.3)
  360 WRITE(IOOUT,808)
      WRITE(IOOUT,872)
  872 FORMAT(57X,28HAERODYNAMIC CLOUD DIMENSIONS)
      WRITE(IOOUT,808)
      WRITE(IOOUT,874)(OBSCOR(I),I=1,2)
  874 FORMAT(41X,28HOBSERVER    COORDINATES      ,2F10.2)
      WRITE(IOOUT,876)HEIGHT
 876  FORMAT(39X,26HTHE HEIGHT OF THE CLOUD IS,10X,F10.2,7H METERS)
      WRITE(IOOUT,878)(CNTRD(IO),IO=1,2)
 878  FORMAT(38X,28HTHE CENTROID COORDINATES ARE,8X,2F10.2)
      WRITE(IOOUT,880)CENWTH
 880  FORMAT(38X,28HTHE WIDTH AT THE CENTROID IS,8X,F10.2, 7H METERS)
      WRITE(IOOUT,882)SPCHT,SPCWTH
 882  FORMAT(39X,12HTHE WIDTH AT,F8.2,11H METERS IS ,5X,F10.2,7H METERS)
      WRITE(IOOUT,884)NCPTS
 884  FORMAT(46X,I3,37H  CONTOUR POINTS HAVE BEEN DETERMINED )
      WRITE(IOOUT,886)((CPTS(IO,IPT),IO=1,2),IPT=1,NCPTS)
  886 FORMAT((60X,2(F10.3,2X)))
  400 CONTINUE
      GO TO 10
```

```
410 WRITE(IOOUT,888)
888 FORMAT(/,38X,48H** THE CLOUD HAS BEEN PRECOMPUTED AND STORED ON,
   1 5H FILE)
    GO TO 10
900 WRITE(IOOUT,901)
901 FORMAT(/,24X,48H*** DIRTRAN ERROR - MORE THAN 15 RECORDS OF DATA,
   1  35H HAVE BEEN INPUT WITHOUT A GO CARD.)
    IERR=1
    GO TO 999
903 WRITE(IOOUT,904)
904 FORMAT(/,39X,47H*** DIRTRAN ERROR - TIMES ARE NOT IN INCREASING,
   + 6H ORDER)
    IERR=1
    GO TO 999
905 WRITE(IOOUT,906)
906 FORMAT(/18X,46H*** DIRTRAN ERROR -NO TRANSMITTER AND RECEIVER,
   + 49H AND/OR OBSERVER COORDINATES HAVE BEEN SPECIFIED.)
    IERR=1
    GO TO 999
909 WRITE(IOOUT,910)
910 FORMAT(/,25X,44H*** DIRTRAN ERROR - ONLY ONE DATA RECORD FOR,
   1 38H CHARGE INFORMATION HAS BEEN SPECIFIED)
    IERR=1
    GO TO 999
911 WRITE(IOOUT,912)
912 FORMAT(/,16X,49H*** DIRTRAN ERROR - MINIMUM AMOUNT OF INFORMATION,
   1 26H REQUIRED IS NOT AVAILABLE,/,10X,14H  CHECK INPUTS)
    IERR=1
    GO TO 999
913 WRITE(IOOUT,914)
914 FORMAT(/20X,49H*** DIRTRAN ERROR - BOTH TRANSMITTER AND RECEIVER,
   1 43H LOCATIONS MUST BE SPECIFIED,  CHECK INPUTS)
    IERR=1
    GO TO 999
915 WRITE(IOOUT,916)
916 FORMAT(/30X,50H IOPT AND NARY DO NOT AGREE SEE THE ABOVE COMMENTS,
   1 21H FOR CORRECT MATCHING)
    IERR=1
999 RETURN
    END
```

```
      SUBROUTINE AMOUNT(VOLSPH,WAKAL,SPHAL)
C
C   SUBROUTINE TO DETERMINE LOADING FOR THE SPHERE AND WAKE
C
C   INPUTS
C
C      VOLSPH - VOLUME OF THE BUOYANTLY RISING SPHERE
C
C      ALL OTHER NEEDED INFORMATION IS PASSED VIA COMMON BLOCKS
C
C   OUTPUTS
C
C      WAKAL  - AMOUNT OF INITIAL LOADING OF SPHERE THAT HAS BEEN
C               DEPOSIITED IN THE WAKE
C
C      SPHAL  - AMOUNT OF INITIAL LOADIING OF SPHERE THAT IS LEFT
C               IN THE SPHERE
C
C
C   FUNCTIONS AND SUBROUTINES NEEDED
C
C      NONE
C
C*****************************************************************
C
      COMMON/NTAL/TNOT,VOLNOT,TNO,CBLEED
      COMMON/BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM
      TSPH=TNO+DELT
      STUFF=CBLEED*(VOLSPH/TSPH-VOLNOT/TNOT)
      WAKAL=AMIN1(SPHNS(1),STUFF)
      SPHAL=SPHNS(1)-WAKAL
      RETURN
      END
```

```
      SUBROUTINE ATMCAL(NATM,ZT,TMES,ZU,UMES,PHI,BETA,DHDT,ERR)
      REAL M,N,K,KM
      LOGICAL ERR,DHDT
      DIMENSION ZT(2),TMES(2),ZU(2),UMES(2),ZL0(6)
      COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,M,N,ZINV
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
      COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT
      COMMON/STARS/USTAR,TSTAR,ZSTAR
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA ZL0/-2.5,-4.5,-13.5,10000.,55.,20./
      DATA OMEGA,K /7.2722E-05,.4/
C  ***********************************************************
C
C
C     PURPOSE
C
C          TO FIT THE BEST POWER-LAW PROFILES OF WINDSPEED AND
C          DIFFUSIVITY CONSISTENT WITH KNOWN RELATIONS GOVERNING
C          THE CONSTANT SHEAR STRESS LAYER TO GIVEN MEASUREMENTS
C          AT ONE OR TWO HEIGHTS.  ALSO TO CALCULATE PARAMETERS
C          NEEDED FOR VERTICAL VARIATION IN WIND DIRECTION LAYER,
C          FOR WIND AND TEMPERATURE PROFILES.
C
C
C     INPUTS
C
C          NATM        INTEGER WHICH IS 0 IF WINDSPEED AND TEMPERATURE
C                      ARE AVAILABLE AT TWO HEIGHTS AND EQUAL TO THE
C                      PASQUILL CATEGORY OTHERWISE.
C
C          ZT          SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS
C                      (IN METERS) AT WHICH TEMPERATURES WILL BE GIVEN.
C                      MUST BE IN ASCENDING ORDER.
C
C          TMES        SINGLY DIMENSIONED ARRAY CONTAINING THE TWO
C                      TEMPERATURE MEASUREMENTS IN DEGREES KELVIN
C                      AT HEIGHTS ZT.
C
C          ZU          SINGLY DIMENSIONED ARRAY CONTAINING ONE OR TWO
C                      HEIGHTS (METERS) AT WHICH WIND SPEEDS WILL BE
C                      GIVEN.  MUST BE IN ASCENDING ORDER.
C
C          UMES        SINGLY DIMENSIONED ARRAY CONTAINING THE ONE OR
C                      TWO WIND SPEED MEASUREMENTS (M/S) AT HEIGHTS UMES.
C
C
C          PHI         LATITUDE OF DETONATION SITE.
C
C          BETA        ANGLE OF WIND VELOCITY VECTOR MEASURED COUNTER-
C                      CLOCKWISE FROM EAST.
C
C          DHDT        A LOGICAL VARIABLE WHICH IS .FALSE. IF THE
C                      INVERSION LAYER HEIGHT IS RELATIVELY CONSTANT
C                      AND .TRUE. IF THE LAYER HEIGHT IS INCREASING.
C
C
C     OUTPUTS
C
C          ERR           A LOGICAL WHICH IS TRUE IF AN ERROR IS INCURRED
C                        DURING THE CALCULATION.
C
C          DXZ0          THE RATIO OF THE DIFFUSIVITY IN THE X DIRECTION
C                        TO THE DIFFUSIVITY IN THE Z DIRECTION. RETURNED
C                        IN COMMON /WNDPRM/.
C
C          DYX0          THE RATIO OF THE DIFFUSIVITY IN THE Y DIRECTION
C                        TO THE DIFFUSIVITY IN THE X DIRECTION. RETURNED
C                        IN COMMON /WNDPRM/.
C
```

```
C         DZ0           THE COEFFICIENT OF Z**N IN THE VERTICAL PROFILE
C                       OF VERTICAL DIFFUSIVITY.  RETURNED IN COMMON
C                       /WNDPRM/.
C
C         U0            THE COEFFICIENT OF Z**M IN THE VERTICAL PROFILE
C                       OF HORIZONTAL WIND SPEED.  RETURNED IN COMMON
C                       /WNDPRM/.
C
C         M             THE EXPONENT OF Z IN THE HORIZONTAL WIND SPEED
C                       PROFILE.  RETURNED IN COMMON /WNDPRM/.
C
C         N             THE EXPONENT OF Z IN THE VERTICAL DIFFUSIVITY
C                       PROFILE.  RETURNED IN COMMON /WNDPRM/.
C
C         ZINV         ESTIMATED INVERSION HEIGHT. RETURNED IN /WNDPRM/.
C
C         USTAR        VELOCITY PROFILE SCALE RETURNED IN COMMON
C                      /STARS/.
C
C         TSTAR        TEMPERATURE PROFILE SCALE. RETURNED IN COMMON
C                      /STARS/.
C
C         ZSTAR        HEIGHT AT WHICH THE VERTICAL VARIATION IN WIND
C                      DIRECTION PROFILES FOR WIND AND TEMPERATURE
C                      TAKE EFFECT.
C
C                     OUTPUT RETURNED IN COMMON /EKWIND/ ARE THE
C                     PARAMETERS NEEDED FOR THE WIND PROFILE ABOVE ZSTAR.
C
C                     OUTPUT RETURNED IN COMMON /EKTEMP/ ARE THE
C                     PARAMETERS NEEDED FOR THE TEMPERATURE PROFILES ABOVE
C                     ZSTAR.
C
C     CALLED FROM DUSTCL
C
C     NEEDED FUNCTIONS AND SUBROUTINES
C
C           TMPCAL      CALCULATES SCALED TEMPERATURE PROFILES
C
C           WNDCAL      CALCULATES SCALED WIND SPEED PROFILES
C
C           DIFFUS     FUNCTION TO CALCULATE THE DIFFUSIVITY AT A GIVEN
C                      HEIGHT.
C
C           TEMP       CALCULATES THE POTENTIAL TEMPERATURE AND GRADIENT
C                      AT A GIVEN HEIGHT.
C
C*******************************************************************
C
      ERR=.FALSE.
C
C     DELTH IS THE DIFFERENCE IN POTENTIAL TEMPERATURE BETWEEN THE
C     TWO HEIGHTS WHERE TEMPERATURE IS GIVEN.
C
      Z0=0.01
      T0=TMES(1)
      IF(NATM.EQ.0)GO TO 100
C
C     ASSIGN ATMOSPHERIC PROFILE ACCORDING TO SPECIFIED PASQUILL
C     CATEGORY
C
C     Z0      FRICTION HEIGHT
C     ZL      MONIN-OBUKOV LENGTH
C     USTAR   THE FRICTION VELOCITY
C     TSTAR   THE SCALING TEMPERATURE
C
      ZL=ZL0(NATM)
      IF(NATM.GE.5)Z0=1.E-04*ABS(ZL)
      IF(NATM.LE.3)Z0=1.E-03*ABS(ZL)
      NP=IFIX(SIGN(1.,ZL))
```

```
      USTAR=UMES(1)/WNDCAL(Z0,ZL,ZU(1))
      TSTAR=TMES(1)*USTAR**2/1.568/ZL
      IF(NATM-4)200,300,210
  100 CONTINUE
C
C     USE ITERATIVE PROCEDURE TO CONVERGE ON BEST ATMOSPHERIC PROFILE
C     TO MATCH DATA AT TWO HEIGHTS
C
      DELTH=TMES(2)-TMES(1)+.0098*(ZT(2)-ZT(1))
      NP=SIGN(1.,DELTH)
      DELU=UMES(2)-UMES(1)
      ZULOG=ALOG(ZU(2)/ZU(1))
      ZTLOG=ALOG(ZT(2)/ZT(1))
      USTAR=(UMES(2)-UMES(1))/ZULOG
      TSTAR=DELTH/ZTLOG
      ZL=.638*TMES(1)*USTAR**2/TSTAR
      IF(ABS(ZL).GE.1000.)GO TO 300
      DO 110 ITER=1,100
      USTAR=DELU/(WNDCAL(Z0,ZL,ZU(2))-WNDCAL(Z0,ZL,ZU(1)))
      TSTAR=DELTH/(TMPCAL(Z0,ZL,ZT(2))-TMPCAL(Z0,ZL,ZT(1)))
      ZLP=ZL
      ZL=.638*TMES(1)*USTAR**2/TSTAR
      IF(ABS((ZL-ZLP)/ZLP).LT..01)GO TO 120
  110 CONTINUE
      ERR=.TRUE.
      GO TO 999
  120 CONTINUE
      IF(ZL.GT.0.)Z0=1.E-04*ABS(ZL)
      IF(ZL.LE.0.)Z0=1.E-03*ABS(ZL)
      IF(NP)200,300,210
  200 CONTINUE
C
C     UNSTABLE ATMOSPHERE
C
      DXZ0=2.6
      M=.079943
      N=4./3.
      DZ0=.7609*USTAR*ABS(ZL)**(1.-N)
      U0=USTAR*14.2478/ABS(ZL)**M
      GO TO 430
  210 CONTINUE
C
C     STABLE ATMOSPHERE
C
      DXZ0=3.3
      N=.45644
      M=.28414
      DZ0=.059517*USTAR*ABS(ZL)**(1.-N)
      U0=USTAR*36.6642/ABS(ZL)**M
      GO TO 430
  300 CONTINUE
C
C     NEUTRAL ATMOSPHERE
C
      DZ0=.4*USTAR
      DXZ0=2.8
      NP=0
      N=1.
      M=1./7.
      U0=45.92*USTAR/ABS(ZL)**M
  430 CONTINUE
C
C     COMMON CALCULATION TO UNSTABLE, NEUTRAL, AND STABLE ATMOSPHERES
C
      DYX0=1.
      IF(NATM.EQ.0)U0=(U0+UMES(2)/ZU(2)**M)/2.
C
C     ESTIMATE THE INVERSION HEIGHT AND COMPUTE THE NECESSARY
C     PARAMETERS FOR THE WIND AND TEMPERATURE PROFILES BETWEEN
C     ZSTAR AND ZINV WHEN DHDT IS .FALSE..
```

```
C
      APHI=PHI*PIRAD
      FREQ=2.*OMEGA*SIN(APHI)
      HC=K*USTAR/FREQ
      IF(ZL.GT.0.0.AND.ZL.LT.1.E3)GO TO 500
      ZINV=HC
      GO TO 501
  500 ZINV=.26*HC
  501 ZSTAR=.13*ZINV
      KM=DIFFUS(Z0,ZL,ZSTAR)
      SP=USTAR*WNDCAL(Z0,ZL,ZSTAR)
      ALP=SQRT(FREQ/(2.*KM))
      IF(DHDT)GO TO 813
      ARG=ALP*ZSTAR
      ARG1=BETA+ALP*ZINV
      ARG2=ALP*(ZINV-ZSTAR)
      C=SP*EXP(ARG)*SIN(ARG1)/SIN(ARG2)
      PYF=C*EXP(-ARG)*COS(ARG)-SP*COS(BETA)
      PXF=C*EXP(-ARG)*SIN(ARG)+SP*SIN(BETA)
      UE=C*EXP(-ARG)*COS(ARG)-PYF
      VE=-C*EXP(-ARG)*SIN(ARG)+PXF
      UHAT=UE/SQRT(UE*UE+VE*VE)
      VHAT=VE/SQRT(UE*UE+VE*VE)
      CALL TEMP(ZSTAR,TA,DTADZ)
      DTADH=0.0
      TC3=(DTADH-DTADZ)/(2.*(ZINV-ZSTAR))
      TC2=DTADZ-2.*TC3*ZSTAR
      TC1=TA-TC2*ZSTAR-TC3*ZSTAR**2
      GO TO 999
  813 ZSTAR=1.E4
  999 RETURN
      END
```

```
      SUBROUTINE AVRG(ZX,TIME,QTOT,XBAVG,SIG2X,SIG2Y)
C
C  ROUTINE FOR FINDING AVERAGES OF THE MOMENTS OF THE DISCS
C
C
C  INPUTS
C
C     ZX    - HEIGHT AT WHICH THE AVERAGES ARE DESIRED
C
C     TIME  - PRESENT TIME
C
C     QTOT  - SUM OF T00*QDSCS
C
C     XBAVG- AVERAGE OF THE FIRST ORDER MOMENTS (ALONGWIND DISPLACEMENT
C            IN THE WIND DIRECTION)
C
C     SIG2X- AVERAGE OF THE SQUARE OF ONE OF THE SECOND ORDER MOMENTS
C            (ALONGWIND SPREAD)
C
C     SIG2Y- AVERAGE OF THE SQUARE OF ANOTHER SECOND ORDER MOMENT
C            (CROSSWIND SPREAD)
C
C*****************************************************************
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
      COMMON/PRTINF/R0,VGRAV(3),NPRTS
      QTOT=0.0
      QSIG2X=0.0
      QSIG2Y=0.0
      QXBAR=0.0
      Z=ZX
      DO 10 I=1,NDSCS
      H=ZDSC(I)
      ROH2=R2DSC(I)
      TOF=TIME-TDSC(I)
      CALL MOMENT(VGRAV(1),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)
      QT=QDSC(I,1)*Q
      QTOT=QTOT+QT
      QSIG2X=QSIG2X+(SIGW2+ROH2/2.)*QT
      QSIG2Y=QSIG2Y+(SIGP2+ROH2/2.)*QT
      QXBAR=QXBAR+(XBAR+XDSC(I))*QT
   10 CONTINUE
      XBAVG=QXBAR/QTOT
      SIG2X=QSIG2X/QTOT
      SIG2Y=QSIG2Y/QTOT
  999 RETURN
      END
```

```
      SUBROUTINE CLIMB(FUNCT,GFUN,P1,FP1,NSERCH,NOCONT)
C
C       THIS MODULE IS A SUBROUTINE THAT FINDS A POINT ON A CONTOUR
C BY FINDING THE GRADIENT VECTOR AT THAT POINT AND MARCHING ALONG
C IT UNTIL IT FINDS ITSELF IN A REGION GREATER THAN THE CONTOUR LEVEL.
C AT WHICH POINT IT MARCHES HORIZONTALLY, HALVING THE STEP SIZE
C UNTIL THE CONTOUR IS REACHED WITHIN SPECIFIED RESOLUTION.
C      IN ADDITION IT WILL DETERMINE IF A CONTOUR EXISTS.
C
C
C     ARGUMENTS PASSED.
C
C     INPUT
C         FUNCT-THE FUNCTION(X,Y) ALSO GIVEN IN EXTERNAL.
C         P1-THE STARTING POINT.
C
C
C    OUTPUT
C
C         P1     - THE POINT ON THE CONTOUR OR THE POINT AT WHICH
C                THE FUNCTION REACHES A MAXIMUM BELOW THE CONTOUR
C                LEVEL
C         FP1    - THE VALUE OF THE FUNCTION AT P
C         NOCONT-THE ERROR FLAG.
C                F-NO PROBLEM
C                T-NO CONTOUR FOUND.
C         ERR-ERROR FLAG RETURNED BY 'NTRSCT'
C              F-NO ERROR
C              T-ITERATION DIVERGED OR MAXIMUM SEARCH AREA EXCEDED
C
C     IN ADDITION,IN COMMON ARE..,
C
C         YMIN-THE LOWER LIMIT ON Y.
C         DELTA- THE STEP SIZE,MODIFIED IN THIS SUBROUTINE.
C         CON-THE CONTOUR LEVEL.
C         RES-THE RESOLUTION LENGTH
C
C     OTHER VARIABLES INCLUDE
C         GRAD-THE GRADIENT VECTOR
C         P0-THE CURRENT POINT ON THE GRADIENT.
C         P1-THE POINT ON THE GRADIENT BEING TESTED
C            TO SEE ABOUT CONTOUR EXISTENCE.
C         FP0,FP1-THE FUNCTION VALUES OF P0 AND P1.
C
C     CALLED SUBROUTINES
C
C         GRAD2-FINDS THE GRADIENT VECTOR OF A FUNCTION AT
C               A POINT AND THE SLOPE THERE.
C         UNIT-CALCULATES THE NORM AND MAGNITUDE OF A 2 VECTOR.
C         VSUM-VECTOR SUM OF THE FORM C=A+SB WHERE S IS SCALAR
C              MULTIPLIER OF B.
C
      EXTERNAL FUNCT
      LOGICAL NOCONT
      DIMENSION GRAD(2),P0(2),P1(2)
      COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
      COMMON/LIMIT/YMIN,FMIN
      NOCONT=.FALSE.
      ONEM=-1.0
      IF (NSERCH.EQ.0)GO TO 7
      DELTA=SIGN(DELTA,FLOAT(NSERCH))
      FP1=FUNCT(P1(1),P1(2))
      IF(FP1.LT.CON)GO TO 25
      GO TO 22
    3 CONTINUE
      P0(1)=P1(1)
      P0(2)=P1(2)
      FP0=FP1
C ** FINDING THE UNIT GRADIENT AND THE NEXT POINT ALONG IT.
    4 CALL GRAD2(P0,FUNCT,RES,GRAD,DFDS)
```

```
    5 CALL VSUM(P0,GRAD,DELTA,P1)
C ** IS THE POINT HEADING BELOW YMIN **
      IF(P1(2).GE.YMIN)GO TO 7
      P1(2)=YMIN
      CALL VSUM(P1,P0,ONEM,GRAD)
      CALL UNIT(GRAD,GRAD,DELTA)
      IF(ABS(DELTA).LT.RES)GO TO 25
    7 FP1=FUNCT(P1(1),P1(2))
C ** HAS THE CONTOUR BEEN CROSSED **
    8 IF(FP1.GE.CON)GO TO 22
      IF(FP1.GT.FP0)GO TO 3
      DELTA=DELTA/2.
      IF(ABS(DELTA).LT.RES)GO TO 25
      GO TO 5
   25 NOCONT=.TRUE.
      GO TO 99
   22 CONTINUE
C     BEGIN HORIZONTAL SEARCH
      P0(2)=P1(2)
   31 P0(1)=P1(1)
      FP0=FP1
   40 P1(1)=P0(1)+DELTA
      FP1=FUNCT(P1(1),P1(2))
      IF(ABS(DELTA).LT.RES/2.)GO TO 99
      IF(FP1.GE.CON)GO TO 31
      DELTA=DELTA/2.
      GO TO 40
   99 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS5,
     1 ERR)
C ***************************************************************
C
C
C     PURPOSE
C
C     CLDIM CALCULATES FIVE CONTOUR POINTS AND CLOUD DIMENSIONS AS
C     SEEN FROM THE SPECIFIED OBSERVER POSITION.  CLDIM REQUIRES CLOUD
C     PARAMETERS FROM THE BUOYANT RISE STAGE OF CLOUD DEVELOPMENT WHICH
C     ARE SUPPLIED IN COMMON STORAGE /BUOYCL/ AND /PRTINF/ AS WELL AS
C     VIEWING GEOMETRY WHICH IS SUPPLIED IN COMMON /GEOM/.  SPCHT IS
C     REQUIRED INPUT IN THE ARGUMENTS.  ALL OUTPUTS ARE ARGUMENTS.
C
C
C     INPUT
C
C     SPCHT     THE SPECIFIED HEIGHT AT WHICH THE WIDTH OF THE CLOUD
C               IS DESIRED. (METERS)
C
C
C
C     OUTPUT
C
C     CNTRD     A SINGLY DIMENSIONED ARRAY OF LENGTH 2 WHICH CONTAINS
C               THE HORIZONTAL AND VERTICAL COORDINATES, RESP., OF THE
C               CLOUD CENTROID.  (METERS)
C     HEIGHT    THE HEIGHT OF THE CLOUD IN METERS
C     CENWTH    THE WIDTH OF THE CLOUD AT THE CENTROID HEIGHT IN METERS
C     SPCWTH    THE WIDTH OF THE CLOUD AT THE SPECIFIED HEIGHT (METERS)
C     NCPTS     THE NUMBER OF CONTOUR POINTS (=6)
C     CPTS      A DOUBLY DIMENSIONED ARRAY OF SIZE (2,N),N.GE.5, WHICH
C               CONTAINS THE HORIZONTAL AND VERTICAL COORDINATES OF
C               THE FIVE CONTOUR POINTS. (METERS)
C
C
C     REQUIRED SUBROUTINES
C
C
C       CLIMB     DETERMINES IF THE CONTOUR EXISTS, IF SO FINDS A POINT
C                 ON THE CONTOUR.
C
C     CALLED BY DUSTCL
C
C ***************************************************************
      DIMENSION CNTRD(2),CPTS5(2,6),TOP(2)
      LOGICAL HORIZ,NOCONT,SWITCH,CHANGE,ERR
      REAL KZ,KX
      COMMON /BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),TIM
      COMMON /PRTINF/R0,VGRAV(3),NPRTS
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON /MODE/ HORIZ
      COMMON /CLOCK/ T,TWIND
      COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF
      COMMON/WNDFRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
      COMMON/TRAN/YTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/SIG/SIG02,SIGC
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
     1 QDSC(20,3)
      COMMON /SPECS/ RES,STEP,TANT,CON
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      EXTERNAL FUNCT,GFUN
      DATA RES,TANT /.4,.1/
      HORIZ=.TRUE.
      ERR=.FALSE.
      CON=ALOG(VISEXT)
      CPTS5(2,1)=SPCHT
      CPTS5(2,6)=SPCHT
      U=U0*CPTS5(2,1)**UM
```

```
      CPTS5(1,1)=T*U*SINTH
      CPTS5(1,6)=CPTS5(1,1)
      NSERCH=-1
      STEP=20.
C
C  CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN ONE DIRECTION AT
C  THE HEIGHT OF THE OBSERVER.
C
      CALL CLIMB(FUNCT,GFUN,CPTS5,FP1,NSERCH,NOCONT)
      NSERCH=1
      STEP=20.
C
C  CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN THE OPPOSITE
C  DIRECTION AT THE HEIGHT OF THE OBSERVER.
C
      CALL CLIMB(FUNCT,GFUN,CPTS5(1,6),FP1,NSERCH,NOCONT)
      SPCWTH=CPTS5(1,6)-CPTS5(1,1)
      NCPTS=6
      IF(T.LE.TWIND)GO TO 50
      CNTRD(1)=(XTR+VTR*(T-TTR))*SINTH+CENDIF
      CNTRD(2)=ZTR
      SIGX2=SIG02+2.*KX*(T-TTR)
      SIGZ2=SIG02+2.*KZ*(T-TTR)
      SIGX=SQRT(SIGX2)
      SIGZ=SQRT(SIGZ2)
      BOT=1./(2.*VISEXT)
      ARG=BOT*QPUFF(1)/PI/SIGX/SIGZ
      IF(ARG.LT.1.0)GO TO 998
      TOP(1)=CNTRD(1)
      TOP(2)=ZTR+SIGZ*SQRT(2.*ALOG(ARG))
      RAD=SIGX*SQRT(2.*ALOG(ARG))
      CENWTH=2.*(RAD+PRJARY)
      HEIGHT=TOP(2)
      GO TO 100
   50 CNTRD(1)=XCM*SCRN(1)+YCM*SCRN(2)+CENDIF
      CNTRD(2)=ZCM
      TOP(1)=XTOP*SCRN(1)+YTOP*SCRN(2)+CENDIF
      TOP(2)=ZCM+RSPH
      IF(TOP(2).GT.ZINV)TOP(2)=ZINV
      HEIGHT=TOP(2)
      CENWTH=2.*(RSPH+PRJARY)
  100  CPTS5(1,2)=CNTRD(1)-CENWTH/2.
      CPTS5(2,2)=CNTRD(2)
      CPTS5(1,3)=TOP(1)-PRJARY
      CPTS5(2,3)=TOP(2)
      CPTS5(1,4)=TOP(1)+PRJARY
      CPTS5(2,4)=TOP(2)
       CPTS5(1,5)=CNTRD(1)+CENWTH/2.
      CPTS5(2,5)=CNTRD(2)
       NCPTS=6
      GO TO 999
  998 WRITE(IOOUT,1000)
 1000 FORMAT(50H *** UPPER PART OF CLOUD HAS DISSIPATED ***        )
  999 RETURN
      END
```

```
      SUBROUTINE COMPCL(NEWATM,NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,
     1                  PHI,NEWSRC,CHWT,NCHRG,NCHS,DETDEP,NSOIL,DSOD,
     2                  NWL,TRNCOR,RECCOR,TIME,DHDT,TRNLOS,NERR)
C
C   CONTROLING ROUTINE FOR PRECOMPUTING A SINGLE , EXPLOSION PRODUCED
C   DUST CLOUD AND STORING IT ON AN EXTERNAL FILE TO BE USED AT A
C   LATER RUNNING OF THE CODE FOR A RANDOM DISTRIBUTION IN SPACE AND TIME
C   OF CHARGES
C
C   INPUTS
C      FOR DETAILS SEE DRTRAN
C
C
C   OUTPUTS
C
C      TRNLOS -CALCULATE TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT
C
C*****************************************************************
      DIMENSION ZTEMP(2),TMPMES(2),ZWND(2),WNDMES(2),TRNCOR(3),
     1RECCOR(3),TRNFRM(2,2),SIDE1(2),SIDE2(2),NCHS(2),ORIG(2),TRN(3),
     2          REC(3),SRCBAS(2),PAS(6)
      LOGICAL DHDT,NEWATM,NEWSRC,ERR,FLAG
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/VL/VLOAD
      COMMON/OPTION/IOPT,IFILE
      COMMON/CLOCK/FTIME,TWIND
      COMMON/BUOYCL/Y(8),SPHNS(3),RTIM
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +           COOR(2,200),TSTAG(200),
     +               ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
     1               RB(3,25),Z2(3,25)
      COMMON/WNDPRM/DXZ0,DYZ0,DZ0,U0,UM,DN,ZINV
      COMMON/CARB/RCARB1,RCARB2
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA FLAG/.TRUE./
      DATA PAS/4HA    ,4HB    ,4HC    ,4HD    ,4HE    ,4HF    /
      IF(IOPT.EQ.4)GO TO 500
C
C   PRECOMPUTE INFORMATION AND STORE ON FORTRAN UNIT IFILE
C
      THETAX=THWND*PI/180.
      TWIND=1.E5
      TTR=1.E5
      IF(.NOT.NEWATM)GO TO 10
      CALL ATMCAL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,PHI,THETAX,DHDT,ERR)
      IF(.NOT.ERR)GO TO 10
      NERR=7
      GO TO 999
   10 CONTINUE
      IF(.NOT.NEWSRC)GO TO 20
      CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD)
   20 CONTINUE
C
C   COMPUTE INITIAL LOADING
C
      SUM=0.0
      DO 25 I=1,NDSCS
      SUM=SUM+QDSC(I,1)
   25 CONTINUE
      VLOAD=SPHNS(1)+SUM
C
C   CALL PRECL TO COMPUTE AND STORE THE QUADRATIC FITS NECESSARY
C   FOR THE CONE
C
      CALL PRECL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,PHI,DHDT,
     1           CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT)
      GO TO 999
  500 CONTINUE
```

```
      IF(.NOT.FLAG) GO TO 35
      READ(IFILE)NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)
      READ(IFILE)DHDT,PHI,CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT,ZINV
      READ(IFILE)VLOAD,RCARB1,RCARB2
      READ(IFILE)ICOUNT
      DO 30 J=1,ICOUNT
      READ(IFILE) TIMES(J),(RT(I,J),RB(I,J),Z2(I,J),XC0(I,J),
     1            XC1(I,J),I=1,3)
   30 CONTINUE
      WRITE(IOOUT,800)
  800 FORMAT(/,5X,85H ATMOSPHERIC, CHARGE, AND SOIL CHARACTERISTICS USED
     1 WHEN THE CLOUD WAS PRECOMPUTED.              )
C
C  WRITE OUT ATMOSPHERIC INFORMATION
C
      WRITE(IOOUT,808)
 808  FORMAT(1X)
      WRITE(IOOUT,810)PAS(NATMOS)
  810 FORMAT(30H  PASQUILL CATEGORY            ,A4)
      NIO=1
      WRITE(IOOUT,812)(ZTEMP(I),TMPMES(I),ZWND(I),WNDMES(I),I=1,NIO)
  812 FORMAT(8H      HT ,F8.2,7H  TEMP ,F8.2,7H     HT,F8.2,7H  WIND ,
     1 F8.2)
      WRITE(IOOUT,814)THWND
  814 FORMAT(22H  WIND DIRECTION       ,F8.2)
      WRITE(IOOUT,819)PHI
      IF(DHDT)WRITE(IOOUT,816)
  816 FORMAT(40H  THE INVERSION LAYER HEIGHT IS GROWING      )
      IF(.NOT.DHDT)WRITE(IOOUT,818)
  818 FORMAT(42H  THE INVERSION LAYER HEIGHT IS CONSTANT     )
  819 FORMAT(/22H  LATITUDE                      ,F8.2)
C
C   WRITE SOIL CHARACTERISTICS
C
      IF(NSOIL.EQ.1)WRITE(IOOUT,821)
  820 FORMAT(15H  SILT CONTENT      ,F5.2)
  821 FORMAT(/,15H  SOIL-1        )
      IF(NSOIL.EQ.2)WRITE(IOOUT,822)
  822 FORMAT(/,15H  SOIL-2        )
      WRITE(IOOUT,820)SILT
      WRITE(IOOUT,823)DSOD
  823 FORMAT(21H  DEPTH OF SOD       ,F5.2)
C
C   WRITE EXPLOSIVE CHARGE CHARACTERISTICS
C
      IF(NCHRG.EQ.1)WRITE(IOOUT,824)
  824 FORMAT(/,65H  SURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC, TIP
     1 ON GROUND        )
      IF(NCHRG.EQ.2)WRITE(IOOUT,825)
  825 FORMAT(/,25H  BARE CHARGE ON SURFACE      )
      IF(NCHRG.EQ.3)WRITE(IOOUT,826)
  826 FORMAT(/,45H  30 DEGREE TILTED TIP AT 0.3 METER DEPTH        )
      IF(NCHRG.EQ.4)WRITE(IOOUT,827)
  827 FORMAT(/,45H  30 DEGREE TILTED TIP AT 0.6 METER DEPTH        )
      IF(NCHRG.EQ.5)WRITE(IOOUT,828)
  828 FORMAT(/,40H  HORIZONTAL PROJECTILE ON SURFACE               )
      WRITE(IOOUT,829)CHWT
  829 FORMAT(30H  WEIGHT OF CHARGE           ,F8.2,5H KG. )
      WRITE(IOOUT,830)DETDEP
  830 FORMAT(30H  DETONATION DEPTH            ,F8.2)
      FLAG=.FALSE.
C
C  COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER
C  DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN
C  THE WIND DIRECTION.
C
 35   CONTINUE
      IF (.NOT.NEWATM) GO TO 45
      THETAX=THWND*PI/180.
      TRNFRM(1,1)=COS(THETAX)
```

```
      TRNFRM(2,2)=TRNFRM(1,1)
      TRNFRM(1,2)=SIN(THETAX)
      TRNFRM(2,1)=-TRNFRM(1,2)
C
C  MAKE THE ORIGIN OF THE LOCAL COORDINAT SYSTEM THE FIRST
C  CHARGE LOCATION THAT WAS INPUT BY THE USER
C
      DO 40 I=1,2
      SRCBAS(I)=COOR(I,1)
      ORIG(I)=SRCBAS(I)
   40 CONTINUE
C
C  CALL SETUP TO COMPUTE THE ARRAY OF DIFFERENCE VECTORS
C
      CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)
C
C  COMPUTE COORDINATES OF THE TRANSMITTER AND RECEIVER IN THE LOCAL
C  COORDINATE SYSTEM
C
      TRN(3)=TRNCOR(3)
      REC(3)=RECCOR(3)
      DO 60 I=1,2
      TRN(I)=0.0
      REC(I)=0.0
      DO 50 J=1,2
      TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))
      REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))
   50 CONTINUE
   60 CONTINUE
45    CONTINUE
C
C  CALL PRETRN TO COMPUTE THE TRANSMITTANCE ALONG THE SPECIFIED LINE OF
C  SIGHT
C
      CALL PRETRN(TRN,REC,TIME,TRNLOS)
  999 RETURN
      END
```

```
      SUBROUTINE CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,
     1XNORM,PLEN)
C
C   ROUTINE TO FIND THE LENGTH OF A NON-HORIZONTAL LINE THAT INTERSECTS
C   A CONE
C
C   INPUTS
C
C      U     - UNIT VECTOR ALONG LINE CONNECTING THE TRANSMITTER AND
C              RECEIVER.
C
C      TR    - TRANSMITTER COORDINATES
C
C      HTTOP- HEIGHT OF THE TOP OF THE CONE SHAPED PORTION OF THE CLOUD
C
C      HTBOT- HEIGHT OF THE BOTTOM OF THE CONE SHAPED PORTION OF THE
C              CLOUD
C
C      XCEN - X POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP
C
C      YCEN - Y POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP
C
C      RTOP - RADIUS OF THE CONE AT THE TOP
C
C      RBOT - RADIUS OF THE CONE AT THE BOTTOM
C
C      XB    - X POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD
C
C      YB    - Y POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD
C
C   OUTPUT
C
C      PLEN - LENGTH OF THE INTERSECTION OF CONE AND THE LINE OF SIGHT
C
C   FUNTIONS AND SUBROUTINES
C
C      NONE
C
C*********************************************************************
      DIMENSION U(3),TR(3)
      IF(U(3).LT.0.0)GO TO 40
C
C   SET UP BOUNDS SO INTERSECTION OF LINE IS SUCH THAT HTBOT < Z < HTTOP
C
      PMIN=(HTBOT-TR(3))/U(3)
      PMAX=(HTTOP-TR(3))/U(3)
      GO TO 50
   40 PMIN=(HTTOP-TR(3))/U(3)
      PMAX=(HTBOT-TR(3))/U(3)
   50 P1=(HTTOP-HTBOT)/U(3)
      P0=(HTBOT-TR(3))/U(3)
C
C   SET UP QUADRATIC TO BE SOLVED
C
      DX1=U(1)*P1-XCEN+XB
      DY1=U(2)*P1-YCEN+YB
      DR=RTOP-RBOT
      A=DX1**2+DY1**2-DR**2
      DX0=TR(1)+U(1)*P0-XB
      DY0=TR(2)+U(2)*P0-YB
      B=2.*(DX1*DX0+DY1*DY0-DR*RBOT)
      C=(DX0**2+DY0**2-RBOT**2)
      RADIC1=(DX1*RBOT-DX0*DR)**2
      RADIC2=(DY1*RBOT-DY0*DR)**2
      RADIC3=(DX1*DY0-DY1*DX0)**2
C
C   DETERMINE PATH LENGTH IF THE LINE INTERSECTS THE CONE
C
      IF(ABS(A).LT.1.E-20)GO TO 60
      RADIC=RADIC1+RADIC2-RADIC3
```

```
      IF(RADIC.LT.0.0)GO TO 68
      ROOT1=-B/A/2.+SQRT(RADIC)/A
      ROOT2=-B/A/2.-SQRT(RADIC)/A
C
C  DETERMINE WHICH POINT IS CLOSEST AND WHICH IS FARTHEST FROM THE
C  TRANSMITTER
C
      TEMP1=P1*ROOT1+P0
      TEMP2=P1*ROOT2+P0
      IF (A.LT.0.0) GO TO 57
      IF(TEMP1.GT.TEMP2)GO TO 55
      PFAR=TEMP2
      PNEAR=TEMP1
      GO TO 67
   55 PFAR=TEMP1
      PNEAR=TEMP2
      GO TO 67
57    IF (U(3).GT.0.0) GO TO 58
      PNEAR=-1.E20
      PFAR=AMIN1(TEMP1,TEMP2)
      GO TO 67
58    PNEAR=AMAX1(TEMP1,TEMP2)
      PFAR=1.E20
      GO TO 67
C
C  CONSIDER DEGENERATE CASE
C
   60 IF(B.LT.1.E-20)GO TO 68
      IF(P1/B.LT.0.0)GO TO 65
      PFAR=P1*(-C/B)+P0
      PNEAR=0.0
      GO TO 67
   65 PFAR=XNORM
      PNEAR=P1*(-C/B)+P0
   67 PLEN=AMIN1(PFAR,XNORM,PMAX)-AMAX1(PNEAR,0.0,PMIN)
      IF(PLEN.LT.0.0)PLEN=0.0
      GO TO 999
   68 PLEN=0.0
  999 RETURN
      END
```

```
      SUBROUTINE CONVRT(T)
      LOGICAL SWITCH,CHANGE
      REAL KZ,KX,N,MDIF
      COMMON/PRTINF/R0,VGRAV(3),NPRTS
      COMMON/BUOYCL/RSPH,DELT,VZSPH,XCMSPH,YCMSPH,ZCMSPH,XTOP,YTOP,
     1SPHNS(3),RISTIM
      COMMON/CLOCK/TIME,TWIND
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/SIG/SIG02,SIGC
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON/LOAD/WAKAL,SPHAL
      COMMON/WNDPRM/DXZ0,DYZ0,DZ0,U0,M,NDIF,ZINV
      COMMON/STARS/USTAR,TSTAR,ZSTAR
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
      COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAVRG
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C ***************************************************************
C
C   PURPOSE
C
C       TO CONVERT THE CURRENT BUOYANT CLOUD INTO A THREE DIMENSIONAL
C       GAUSSIAN PUFF TO BE USED BY THE WIND DISPERSION MODEL.
C
C   INPUT
C
C     T    TIME IN SECONDS AFTER DETONATION
C
C
C   CALLED BY RISE
C
C      FUNCTIONS AND SUBROUTINES NEEDED
C
C        WNDCAL
C        AMOUNT
C        AVRG
C
C****************************************************************
      CHANGE=.TRUE.
      TWIND=T
      VTR=USTAR*WNDCAL(Z0,ZL,ZCMSPH)
      KZ=DIFFUS(Z0,ZL,ZCMSPH)
      KX=KZ*DXZ0
      TTR=T
      XTR=XCMSPH
      ZTR=ZCMSPH
      DO 10 IPRTS=1,NPRTS
      QPUFF(IPRTS)=SPHNS(IPRTS)
   10 CONTINUE
      VOLSPH=(4./3.)*PI*RSPH**3
      CALL AMOUNT(VOLSPH,WAKAL,SPHAL)
      QPUFF(1)=SPHAL
   60 SIG02=((2./3.)*RSPH)**2
      SIG0=SQRT(SIG02)
      ZX=5.0
   61 CALL AVRG(ZX,T,QTOT,XBAVRG SIG2X,SIG2Y)
      SIGX=SQRT(SIG2X)
      SIGY=SQRT(SIG2Y)
C
C   COMPUTE PARAMETERS NEEDED FOR LOCAL AND COLUMN DENISTY FOR THE WAKE
C   SOLUTION
C
      SIGC=(SQRT(SIGX*SIGY)+SQRT(SIG02))/2.
      XDIF=XCMSPH-XBAVRG
      YDIF=YCMSPH
      ZDIF=ZCMSPH-5.0
      TDIF=SIGC**2/KX/2.
      TDX=2.*KX
      TDZ=2.*KZ
      QLOC=WAKAL/4./PI/SQRT(2.)
```

```
      QCOL=WAKAL/4./SQRT(PI)                                            COV00700
  998 RETURN                                                            COV00710
      END                                                               COV00720
```

```
      FUNCTION CSPHER(X,Y,Z,T)
C
C   COMPUTE EITHER THE COLUMN DENSITY FOR A GIVEN LINE OF SIGHT OR
C   COMPUTE THE LOCAL CONCENTRATION AT X,Y,Z FOR THE SPHERE
C
C   INPUT
C
C       X,Y,Z     COORDINATES IN METERS.  IF LINE INTEGRAL IS DESIRED,
C                 Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.
C
C       T         THE TIME IN SECONDS AFTER DETONATON
C
C   OUTPUT
C
C      RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND
C      THE LINE INTEGRAL OF CONCENTRATION (COLUMN DENSITY) IF HORIZ
C      IS .TRUE.
C
C   FUNTIONS AND SUBROUTINES CALLED
C
C       NONE
C
C***********************************************************************
      REAL M,N,KZ,KX
      LOGICAL HORIZ,SWITCH,CHANGE,TEST
      COMMON/PRTINF/ R0,VGRAV(3),NPRTS
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON /MODE/ HORIZ
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
     1 QDSC(20,3)
      COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +              COOR(2,200),TSTAG(200),DMMY(401)
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/SIG/SIG02,SIGC
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/LOAD/WAKAL,SPHAL
      COMMON/ACL/CWINDS,CWINDC,CWINDW
      COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAVRG
      COMMON/LOS/TR(3),RE(3),U(3)
      COMMON/CHARGE/NCHG
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      CSPHER=0.0
      CWNDSC=0.0
      CWINDC=0.0
      IF(NARY.EQ.3)GO TO 999
      IF((T-TTR).LT.1.E-20)GO TO 999
C
C   COMPUTE CONTRIBUTION FROM BUOYANT CLOUD AFTER SWITCHING TO THE
C   WIND MODEL USING A THREE DIMENSIONAL GAUSSIAN PUFF
C
      SIGX2=SIG02+2.*KX*(T-TTR)
      SIGZ2=SIG02+2.*KZ*(T-TTR)
      SIGX=SQRT(SIGX2)
      SIGZ=SQRT(SIGZ2)
      ARG2=(Z-ZTR)**2/(2.*SIGZ2)
      IF(ABS(ARG2).GT.30.)GO TO 25
      TERM2=EXP(-ARG2)
      GO TO 26
   25 TERM2=0.0
   26 ARG3=(Z+ZTR)**2/(2.*SIGZ2)
      IF(ABS(ARG3).GT.30.)GO TO 27
      TERM3=EXP(-ARG3)
      GO TO 28
   27 TERM3=0.0
   28 IF(HORIZ)GO TO 50
C
C   COMPUTE CONCENTRATION AT X,Y,Z
C
```

```
      XC=XTR+VTR*(T-TTR)+DIFF(1,NCHG)
      YC=DIFF(2,NCHG)
      ARG1=((X-XC)**2+(Y-YC)**2)/2./SIGX2
      IF(ABS(ARG1).GT.30.)GO TO 999
      TERM=SPHAL/((2.*PI)**(3./2.))/SIGZ/SIGX2
      CWINDC=TERM*EXP(-ARG1)*(TERM2+TERM3)
      GO TO 100
C
C  COMPUTE COLUMN DENSITY
C
   50 TERM=SPHAL/2./PI/SIGX/SIGZ
      DO 90 I=1,NCHTOT
      XC=XTR+VTR*(T-TTR)+DIFF(1,I)
      YC=DIFF(2,I)
      XY=XC*SCRN(1)+YC*SCRN(2)
      ARG1=(X-XY)**2/2./SIGX2
      IF(ABS(ARG1).GT.30.)GO TO 90
      CWNDSC=TERM*EXP(-ARG1)*(TERM2+TERM3)
      CWINDC=CWINDC+CWNDSC
      CALL TRNCHK(CWINDS,CWINDW,CWINDC)
      IF(TEST)GO TO 999
   90 CONTINUE
  100  CSPHER=CWINDC
  999 RETURN
      END
```

```
      FUNCTION CWAKE(X,Y,Z,T)
C
C   FUNCTION TO COMPUTE THE LOCAL CONCENTRATION OR COLUMN DENSITY AT
C   X,Y,Z AND TIME T AFTER THE BLAST FOR THE WAKE.
C
C
C   INPUT
C
C      X,Y,Z       COORDINATES IN METERS. IF THE LINE INTEGRAL IS DESIRED
C                  THESE ARE NOT USED AND THE LINE IS SPECIFIED BY THE
C                  TRANSMITTER AND RECEIVER COORDINATES AND INFORMATION
C                  CALCULATED AT THE TIME THE BUOYANT FIREBALL CONVERTED
C                  TO THE WIND MODEL.
C
C      T           THE TIME IN SECONDS AFTER DETONATION
C
C
C   OUTPUT
C
C      RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND
C      THE LINE INTEGRAL OF CONCENTRATION  (COLUMN DENSITY) IF
C      HORIZ IS .TRUE.
C
C
C   FUNCTIONS AND SUBROUTINES NEEDED
C
C      ERF     COMPUTE THE ERROR FUNCTION
C
C*********************************************************************
      REAL M,N,KZ,KX
      LOGICAL HORIZ,SWITCH,CHANGE,TEST
      COMMON/PRTINF/ R0,VGRAV(3),NPRTS
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON /MODE/ HORIZ
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
     1 QDSC(20,3)
      COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +             COOR(2,200),TSTAG(200),DMMY(401)
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/SIG/SIG02,SIGC
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/LOAD/WAKAL,SPHAL
      COMMON/ACL/CWINDS,CWINDC,CWINDW
      COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAVRG
      COMMON/LOS/TR(3),RE(3),U(3)
      COMMON/CHARGE/NCHG
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      IF(NARY.EQ.3)GO TO 999
      IF((T-TTR).LT.1.E-20)GO TO 999
C
C   COMPUTE CONTRIBUTION FROM THE WAKE AFTER SWITCHING TO THE WIND MODEL
C   FOR A SINGLE CHARGE
C
      CWAKE=0.0
      CWINDW=0.0
      CWNDSW=0.0
      SIGX2=TDX*(TDIF+(T-TTR))
      SIGZ2=TDZ*(T-TTR)
      IF(HORIZ)GO TO 210
      XB=XBAVRG+DIFF(1,NCHG)
      YB=DIFF(2,NCHG)
C
C   COMPUTE THE LOCAL CONCENTRATION
C
      A=-(XDIF**2+YDIF**2)/2./SIGX2-(ZDIF**2/2./SIGZ2)
      XX=X-VTR*(T-TTR)
      B0=(XDIF*(XX-XB)+YDIF*(Y-YB))/SIGX2
      B1=B0+(ZDIF*(Z-5.)/SIGZ2)
```

```
      B2=B0-(ZDIF*(Z+5.)/SIGZ2)
      C0=-((XX-XB)**2+(Y-YB)**2)/2./SIGX2
      C1=C0-((Z-5.)**2/2./SIGZ2)
      C2=C0-((Z+5.)**2/2./SIGZ2)
      RTMA=SQRT(-A)
      ARG1=C1-(B1**2/4./A)
      ARG2=-B1/2./RTMA
      ARG3=RTMA+ARG2
      ARG4=C2-(B2**2/4./A)
      ARG5=-B2/2./RTMA
      ARG6=RTMA+ARG5
      IF(ARG1.LT.-30.)GO TO 221
      TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))
      GO TO 222
  221 TERM1=0.0
  222 CONTINUE
      IF(ARG4.LT.-30.)GO TO 223
      TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))
      GO TO 224
  223 TERM2=0.0
  224 CWAKE=QLOC/SIGX2/SQRT(SIGZ2)/RTMA*(TERM1+TERM2)
      GO TO 999
C
C  COMPUTE COLUMN DENSITY
C
  210 DO 245 J=1,NCHTOT
      XB=XBAVRG+DIFF(1,J)
      YB=DIFF(2,J)
  230 A=-(XDIF*U(2)-YDIF*U(1))**2/2./SIGX2-(ZDIF)**2/2./SIGZ2
      TRR=TR(1)-VTR*(T-TTR)
      B0=(YDIF*U(1)-XDIF*U(2))*((TR(2)-YB)*U(1)-(TRR-XB)*U(2))/SIGX2
      B1=B0+ZDIF*(TR(3)-5.)/SIGZ2
      B2=B0-ZDIF*(TR(3)+5.)/SIGZ2
      C0=-((TR(2)-YB)*U(1)-(TRR-XB)*U(2))**2/2./SIGX2
      C1=C0-(TR(3)-5.)**2/2./SIGZ2
      C2=C0-(TR(3)+5.)**2/2./SIGZ2
      RTMA=SQRT(-A)
      ARG1=C1-B1**2/4./A
      ARG2=-B1/2./RTMA
      ARG3=RTMA+ARG2
      ARG4=C2-B2**2/4./A
      ARG5=-B2/2./RTMA
      ARG6=RTMA+ARG5
      IF(ARG1.GT.30.)ARG1=30.
      IF(ARG1.LT.-30.)GO TO 231
      TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))
      GO TO 232
  231 TERM1=0.0
  232 CONTINUE
      IF(ARG4.LT.-30.)GO TO 233
      TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))
      GO TO 234
  233 TERM2=0.0
  234 ARG=SIGX2*S'GZ2
      CWNDSW=(QCOL/SQRT(ARG)/RTMA)*(TERM1+TERM2)
  240 CONTINUE
      CWINDW=CWINDW+CWNDSW
      CALLTRNCHK(CWINDS,CWINDW,CWINDC)
      IF(TEST)GO TO 999
  245 CONTINUE
      CWAKE=CWINDW
  999 RETURN
      END
```

```
      FUNCTION CWIND(X,Y,Z,T)
C
C     PURPOSE
C
C       TO COMPUTE THE CONCENTRATION AT A POINT OR INTEGRATED ALONG
C       A HORIZONTAL LINE  (SKIRT PORTION OF DUST CLOUD)
C
C     INPUT
C
C       X,Y,Z     COORDINATES IN METERS.  IF LINE INTEGRAL IS DESIRED,
C                 Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.
C
C       T         THE TIME IN SECONDS AFTER DETONATION
C
C     OUTPUT
C
C       RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND
C       THE LINE INTEGRAL OF CONCENTRATION IF HORIZ IS .TRUE.
C
C     SUBROUTINES CALLED
C
C       MOMENT    COMPUTES ZERO ORDER MOMENT AND INTERPOLATES FROM
C                 TABLE OF HIGHER ORDER MOMENTS.
C
C       CALLED BY FUNCT,TRNCAL
C
C*****************************************************************
      REAL M,N,KZ,KX
      LOGICAL HORIZ,SWITCH,CHANGE,TEST,SKIP
      DIMENSION REF(2),REF0(2)
      COMMON/PRTINF/ R0,VGRAV(3),NPRTS
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON /MODE/ HORIZ
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
     1 QDSC(20,3)
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +             COOR(2,200),TSTAG(200),DMMY(401)
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/SIG/SIG02,SIGC
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/LOAD/WAKAL,SPHAL
      COMMON/ACL/CWINDS,CWINDC,CWINDW
      COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAVRG
      COMMON/LOS/TR(3),RE(3),U(3)
      COMMON/CHARGE/NCHG
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
      COMMON/SKIPIT/SKIP
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C
C    COMMON /PRTINF/
C
C      R0      INITIAL RADIUS OF THE CLOUD IN METERS
C      VGRAV   SINGLY DIMENSIONED ARRAY.  VGRAV(I) IS THE OPTICALLY
C              WEIGHTED AVERAGE SETTLING VELOCITY FOR PARTICLES IN THE
C              I SIZE RANGE
C      NPRTS   THE NUMBER OF PARTICLE SIZE RANGES
C
C    COMMON /DISCS/
C
C      NDSCS   THE NUMBER OF DISC SOURCES
C      TDSC    SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF RELEASE
C              OF THE DISC SOURCES
C      XDSC    SINGLY DIMENSIONED ARRAY CONTAINING THE X COORDINATE
C              OF THE CENTER OF THE DISC SOURCES
C      ZDSC    SINGLY DIMENSIONED ARRAY CONTAINING THE Z COORDINATE
C              OF THE CENTER OF THE DISC SOURCES
C      R2DSC   SINGLY DIMENSIONED ARRAY CONTAINING THE SQUARE OF THE
C              RADII OF THE DISC SOURCES
C      QDSC    DOUBLY DIMENSIONED ARRAY.   QDSC(I,J) IS THE OPTICALLY
```

```
C              WEIGHTED MASS OF PARTICLES OF THE J SIZE RANGE IN THE
C              I DISC.
C
C
C     SUM THE CONTRIBUTIONS OF THE DISC SOURCES TO THE
C     OPTICALLY WEIGHTED CONCENTRATION AT (X,Y,Z,T)
C
      CWIND=0.
      CWNDSC=0.0
      CWINDS=0.0
      CWINDC=0.0
      CWINDW=0.0
      IF(HORIZ)GO TO 110
C
C   COMPUTE CONCENTRATION AT X,Y,Z  (FOR SIMULTANEOUS BURST)
C
      DO 100 I=1,NDSCS
      REF0(1)=XDSC(I)
      REF0(2)=0.0
      ROH2=R2DSC(I)
      H=ZDSC(I)
      TOF=T-TDSC(I)
      DO 90 J=1,NPRTS
C
C   DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z
C
      CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)
      IF(Q.GT.1.E-10)GO TO 50
      CWNDSC=0.0
      GO TO 100
   50 CONTINUE
      RX2=ROH2+2.*SIGW2
      RY2=ROH2+2.*SIGP2
      DO 114 NA=1,2
      REF(NA)=REF0(NA)+DIFF(NA,NCHG)
  114 CONTINUE
      ARG=-(X-REF(1)-XBAR)**2/RX2
      IF(ABS(ARG).GT.30.)GO TO 100
      CWNDSC=(Q/RTPI/SQRT(RX2))*EXP(ARG)
      ARG=-(Y-REF(2))**2/RY2
      IF(ABS(ARG).GT.30.)GO TO 100
      CY=EXP(ARG        )/RTPI/SQRT(RY2)
      CWNDSC=QDSC(I,J)*CWNDSC*CY
      CWINDS=CWINDS+CWNDSC
   90 CONTINUE
  100 CONTINUE
      CWIND=CWINDS
      GO TO 999
  110 DO 220 ICHG=1,NTOT
      IF(T.LT.TSTAG(ICHG))GO TO 220
      DO 211 I=1,NDSCS
      TOF=T-TDSC(I)-TSTAG(ICHG)
      REF0(1)=XDSC(I)
      REF0(2)=0.
      ROH2=R2DSC(I)
      H=ZDSC(I)
      IF(HORIZ) REF0(1)=REF0(1)*SINTH
      DO 210 J=1,NPRTS
      CWNDSC=0.0
C
C     DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z
C
      CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)
      IF(Q.GT.1.E-10) GO TO 113
C
C     IF Q IS TOO SMALL, ITS CONTRIBUTION IS IGNORED
C
      CWNDSC=0.
      GO TO 210
  113 CONTINUE
```

```
      DO 200 MA=1,NCHTOT
      INDX=MA
      IF(NARY.EQ.3)INDX=ICHG
      RX2=R0H2+2.*SIGW2
      RY2=R0H2+2.*SIGP2
C
C     COMPUTE CONCENTRATION ALONG LINE OF SIGHT SPECIFIED BY X,Z
C
  120 CONTINUE
      REF(1)=REF0(1)+PRSEP(INDX)
      REFF2=RX2*SINTH2+RY2*COSTH2
      ARG=-(X-REF(1)-XBAR*SINTH)**2/REFF2
      IF(ABS(ARG).GT.30.)GO TO 150
      CWNDSC=EXP(ARG)           /SQRT(REFF2)/RTPI
      CWNDSC=CWNDSC*Q*QDSC(I,J)
  150 CONTINUE
      CWINDS=CWINDS+CWNDSC
      IF(SKIP)GO TO 190
      CALL TRNCHK(CWINDS,CWINDW,CWINDC)
      IF(TEST)GO TO 999
  190 CONTINUE
  200 CONTINUE
  210 CONTINUE
  211 CONTINUE
  220 CONTINUE
      CWIND=CWINDS
  999 RETURN
      END
```

```
      SUBROUTINE DIFEQ(N,T,Y,YP)
      REAL KM,KZ,KX
      LOGICAL SWITCH,CHANGE
      DIMENSION Y(N),YP(N)
      COMMON/PRTINF/ROCL,VGRAV(3),NPRTS
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
      COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +            DMM(600),DMMY(401)
      COMMON /BURST/ ACCEL,TBURST
      COMMON/STARS/USTAR,TSTAR,ZSTAR
      COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/SIG/SIG02,SIGC
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DATA ALPHAK/.25/
C ***************************************************************
C
C
C     PURPOSE
C
C     DIFEQ CONTAINS THE PARTIAL DIFFERENTIAL EQUATIONS FOR THE
C     RISE OF A BUOYANT CLOUD WHICH ARE USED BY SUBROUTINE RKM.
C
C
C     INPUT
C
C     N            THE NUMBER OF DEPENDENT VARIABLES
C     T            THE INDEPENDENT VARIABLE, I.E. TIME
C    Y(1)          RADIUS OF CLOUD
C    Y(2)          CLOUD TEMPERATURE MINUS SURROUNDING TEMPERATURE
C    Y(3)          VERTICAL VELOCITY OF CLOUD
C    Y(4)          X-COORDINATE OF CENTER OF MASS FOR THE CLOUD
C    Y(5)          Y-COORDINATE OF CENTER OF MASS FOR THE CLOUD
C    Y(6)          THE HEIGHT OF THE CLOUD C.O.M.
C    Y(7)          X-COORDINATE OF TOP OF CLOUD
C    Y(8)          Y-COORDINATE OF TOP OF CLOUD
C
C
C     OUTPUT
C
C     YP   AN ARRAY CONTAINING COMPUTED DERIVATIVES OF THE DEPENDENT
C          VARIABLES WITH RESPECT TO THE INDEPENDENT VARIABLE.
C
C
C
C     REQUIRED FUNCTIONS
C
C       TEMP     CALCULATES AMBIENT ATMOSPHERIC TEMPERATURE AND THE
C                TEMPERATURE GRADIENT AT CLOUD HEIGHT.
C
C       WIN      CALCULATES THE WIND SPEED IN THE X AND Y DIRECTION AT
C                CLOUD HEIGHT.
C
C       DIFFUS   COMPUTES THE DIFFUSIVITY AT A SPECIFIED HEIGHT.
C
C     CALLED BY RKM
C
C ***************************************************************
      IF(T.LT.TBURST)GO TO 200
C
      IF(Y(6).GT.ZSTAR)GO TO 5
      CALL TEMP(Y(6),TA,DTADZ)
      GO TO 6
    5 TA=TC1+TC2*Y(6)+TC3*Y(6)**2
      DTADZ=TC2+2.*TC3*Y(6)
    6 CALL WIN(Y(6),XWCM,YWCM)
      TOP=Y(6)+Y(1)
      CALL WIN(TOP,XWTOP,YWTOP)
```

```
      TDIF=TA-T0
C
C     TA        THE AMBIENT ATMOSPHERIC TEMPERATURE AT CLOUD HEIGHT
C     DTADZ     THE TEMPERATURE GRADIENT AT CLOUD HEIGHT
C     XWCM      THE WIND SPEED IN THE X DIRECTION AT CLOUD C.O.M.
C     YXCM      THE WIND SPEED IN THE Y DIRECTION AT CLOUD C.O.M.
C     XWTOP     THE WIND SPEED IN THE X DIRECTION AT THE TOP OF THE CLOU
C     YWTOP     THE WIND SPEED IN THE Y DIRECTION AT THE TOP OF THE CLOU
C     TR        THE RATIO OF CLOUD TEMPERATURE TO AMBIENT TEMPERATURE
C
      TR=Y(2)/TA
C
C     CALCULATE ARVOL, THE SURFACE AREA TO VOLUME RATIO
C
      ARVOL=3./Y(1)
C
C  DEFINITION OF DIFFERENTIAL EQUATIONS
C
      YP(1)=ALPHAK*ABS(Y(3))
      ZZ1=Y(6)
      KM=DIFFUS(Z0,ZL,ZZ1)
      GROWTH=KM/Y(1)
      IF(YP(1).LT.GROWTH)YP(1)=GROWTH
      YP(2)=-(1.+TR)*ARVOL*Y(2)*YP(1)-Y(3)*(DTADZ)
      YP(3)=9.8*TR-1.4*ARVOL*Y(3)*YP(1)
      IF(Y(1)+Y(6).GT.ZINV)YP(3)=0.0
      YP(4)=XWCM
      YP(5)=YWCM
      YP(6)=Y(3)
      IF(Y(1)+Y(6).GT.ZINV)YP(6)=0.0
      YP(7)=XWTOP
      YP(8)=YWTOP
      GO TO 999
  200 CONTINUE
      DO 210 I=1,N
      YP(I)=0.
  210 CONTINUE
      YP(3)=ACCEL
      YP(6)=Y(3)
  999 RETURN
      END
```

```
      FUNCTION DIFFUS(Z0,ZL,Z)
      COMMON/STARS/USTAR,TSTAR,ZSTAR
      COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT
C
C ***************************************************************
C  PURPOSE
C
C     TO CALCULATE THE DIFFUSIVITY AT A GIVEN HEIGHT
C
C  INPUTS
C
C     Z0   FRICTION HEIGHT IN METERS.
C
C     ZL   MONIN OBUKHOV LENGTH IN METERS.
C
C     Z    HEIGHT AT WHICH DIFFUSIVITY IS DESIRED.
C
C     CALLED BY ATMCAL, RISE AND DIFEQ
C
C     SUBROUTINES AND FUNCTIONS NEEDED
C
C       NONE
C
C****************************************************************
      ZZ=Z
      IF(Z.GT.ZSTAR)Z=ZSTAR
C
C      NEUTRAL CASE
C
      IF(ABS(ZL).LT.1.E3)GO TO 100
      DIFFUS=.4*USTAR*Z
      GO TO 999
  100 IF(ZL.GT.0.0)GO TO 200
C
C     UNSTABLE CASE
C
      S=Z/ZL
      IF(S.LT.-2.)GO TO 110
      DIFFUS=.4*ABS(ZL*USTAR*S*(1.-16.*S)**(1./4.))
      GO TO 999
  110 DIFFUS=.4*ABS(ZL*(3./AW)*(-1.*S)**(4./3.))*USTAR
      GO TO 999
C
C     STABLE CASE
C
  200 S=Z/ZL
      S0=Z0/ZL
      IF(S.GT.1.5)GO TO 210
      DIFFUS=.4*ZL*USTAR*ABS(1./(1./(S0+S)+7.))
      GO TO 999
  210 DIFFUS=.4*ZL*USTAR*ABS(1./BW)
      Z=ZZ
  999 RETURN
      END
```

```
      FUNCTION DOTPRD(A,B)                                              DOT00010
      DIMENSION A(2),B(2)                                               DOT00020
C                                                                       DOT00030
C     DOTPRD IS THE SCALER PRODUCT OF A AND B                           DOT00040
C                                                                       DOT00050
      DOTPRD=A(1)*B(1)+A(2)*B(2)                                        DOT00060
      RETURN                                                            DOT00070
      END                                                               DOT00080
```

```
      FUNCTION DTERPI(NDIM,XI,XVAL,VAL,VMIN,WORK)
C  **************************************************
C
C
C     PURPOSE
C
C PERFORMS AN N-DIMENSIONAL LINEAR INTERPOLATION
C
C
C     INPUT
C
C    NDIM - THE NUMBER OF DIMENSIONS.  (- DONT RECALCUALTE WEIGHTS)
C      XI - THE POINT IN THE HYPERSPACE AT WHICH THE INTERPOLATED
C           VALUE IS DESIRED.  XI MUST BE A VECTOR OF ATLEAST NDIM
C           IN LENGTH.
C    XVAL - THE COORDINATE VALUES AT THE CORNERS OF THE HYPERCUBE.
C           THE VECTOR MUST BE SET UP LIKE A TWO-DIMENSIONAL ARRAY
C           (2 X NDIM), WHERE THE FIRST SUBSCRIPT REFERS TO THE
C           HYPERCUBE COORDINATES IN THE SECOND SUBSCRIPTS
C           DIRECTION.
C     VAL - THE FUNCTIONAL VALUES AT THE CORNERS OF THE HYPERCUBE
C           SURROUNDING XI.  THIS VECTOR MUST BE FILLED EQUIVALENT
C           TO AN NDIM ARRAY WITH EACH DIMENSION AS 2.  THE SIZE
C           OF VAL SHOULD BE ATLEAST 2**NDIM.
C    VMIN - A MINIMUM VALUE OF VAL FOR WHICH THE INTERPOLATION
C           WILL USE A CORNER VALUE.
C    WORK - A WORK VECTOR OF ATLEAST NDIM*2.  USE TO STORE COOR-
C           DINATE WEIGHTS.
C
C
C     OUTPUT
C
C       RETURNS INTERPOLATED VALUE OF VAL AT XI
C
C     CALLED BY MOMENT
C
C  **********************************************************
      DIMENSION XI(4),XVAL(8),VAL(16),WORK(8)
C
C SET UP THE COORDINATE WEIGHTS
C
      NDI=IABS(NDIM)
      IF(NDIM .LT. 0) GO TO 1
      DO 100 I=1,NDI
      I2=I*2
      I1=I2-1
      WORK(I2)=(XI(I)-XVAL(I1))/(XVAL(I2)-XVAL(I1))
      WORK(I1)=1. - WORK(I2)
100   CONTINUE
C
C INTERPOLATE  -  USE BINARY COUNTER FOR COORDINATE LOCATION
C
1     DTERPI=0.
      SUM=0.
      ND=2**NDI
      DO 201 I=1,ND
      IF(VAL(I) .LT. VMIN) GO TO 201
      L=I-1
      WEIGHT=1.
      DO 200 J=1,NDI
      N=MOD(L,2) + J*2 - 1
      WEIGHT=WEIGHT*WORK(N)
      L=L/2
200   CONTINUE
      SUM=SUM + WEIGHT
      DTERPI=DTERPI + WEIGHT*VAL(I)
201   CONTINUE
      IF(SUM .EQ. 0.) GO TO 202
      DTERPI=DTERPI/SUM
      RETURN
```

```
C                                                                      DTI00710
202   STOP                                                             DTI00720
      END                                                              DTI00730
```

```
      SUBROUTINE DTERPS(II,X,VAL,NZ)
      DIMENSION X(81,4,3),VAL(16),II(4)
C  ***********************************************************
C
C
C     PURPOSE
C
C       TO SET UP A ONE DIMENSIONAL ARRAY OF THE VALUES CORRESPONDING
C       TO THE CORNERS OF THE CUBE WITHIN A TABULATED ARRAY WITH
C       LOWEST CORNER INDICES GIVEN
C
C     INPUT
C
C       II       SINGLY DIMENSIONED ARRAY CONTAINING THE INDICES OF THE
C                LOWEST CORNER OF THE CUBE
C       X        A TRIPLY DIMENSIONED ARRAY CONTAINING THE TABULATED
C                VALUES TO BE SET UP.  THE FIRST INDEX IS THE COLLAPSED
C                INDEX FOR THE FIRST TWO INDICES OF A FOUR-DIMENSIONAL
C                ARRAY
C       NZ       THE RANGE OF THE FIRST INDEX OF THE FOUR-DIMENSIONAL
C                ARRAY
C
C
C     OUTPUT
C
C       VAL      SINGLY DIMENSIONED ARRAY CONTAINING THE VALUES OF X
C                FOR THE 16 CORNER POINTS OF THE CUBE
C
C     CALLED BY MOMENT
C
C  *****************************************************************
      M=0
      DO 104 L=1,2
      LX=L + II(4) - 1
      DO 103 K=1,2
      KX=K + II(3) - 1
      DO 102 JI=1,2
      JIX=(JI + II(2) - 2)*NZ
      DO 101 IJ=1,2
      IJX=JIX + IJ + II(1) -1
      M=M+1
      VAL(M)=X(IJX,KX,LX)
101   CONTINUE
102   CONTINUE
103   CONTINUE
104   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DUSTCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,
     1  THWND,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,
     2  LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,
     3  TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS,NERR,
     4  NCHS,SRCBAS,SIDE1,SIDE2,DHDT)
      LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM,HORIZ,ERR
      LOGICAL SWITCH,CHANGE,DHDT
      DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2),TRNCOR(3)
     1   ,RECCOR(3),CPTS(2,6),ORIG(2),TRNFRM(2,2),TRN(3),REC(3)
     2                 ,CNTRD(2),OBSCOR(2),DIR(2)
     3   ,SRCBAS(2),SIDE1(2),SIDE2(2),TEMP(2),NCHS(2)
      REAL KZ,KX
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON /MODE/ HORIZ
      COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
      COMMON /CLOCK/ FTIME,TWIND
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +             COOR(2,200),TSTAG(200),DMMY(401)
      COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/CARB/RCARB1,RCARB2
      COMMON/ACL/CWINDS,CWINDC,CWINDW
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA ONEM/-1./
C ********************************************************************
C
C
C     PURPOSE
C
C          DUSTCL CALCULATES DUST CLOUD DIMENSIONS AND TRANSMITTANCES
C     THROUGH DUST CLOUDS FOR GIVEN METEOROLOGICAL DATA, SOIL TYPE,
C     EXPLOSIVE CHARACTERISTICS, AND WAVELENGTH.
C
C     SEE COMMENTS IN DRTRAN FOR DETAILS.
C
C     SUBROUTINES CALLED
C
C     ATMCAL      ACCEPTS METEOROLOGICAL DATA AS ARGUMENTS AND COMPUTES
C                 NECESSARY PARAMETERS IN COMMON /WNDPRM/, /EKWIND/
C                 /EKTEMP, /STARS/
C
C     SOURCE      ACCEPTS SOIL, CHARGE, AND WAVELENGTH SPECIFICATIONS
C                 AS INPUT AND COMPUTES NECESSARY PARAMETERS AND INITIAL
C                 VALUES IN COMMON /PRTINF/ , /BUOYCL/ AND /CARB/
C
C     SETUP       ACCEPTS THE USER DEFINED COORDINATES OF THE CHARGES
C                 AND CONVERTS THEM TO THE INTERNAL (LOCAL) CORRDINATE
C                 SYSTEM. ALSO COMPUTES THE AREA AND PERIMETER OF THE
C                 BOUNDING PARALLELOGRAM AND OVERLAP DISTANCE OF THE
C                 CHARGES AND RETURNS THEM IN COMMON /ARRAY/ AND
C                 /SEPRTN/.
C
C     RISE        GIVEN CLOUD DIMENSIONS DURING BUOYANT RISE DEVELOPMENT
C                 OF CLOUD, RISE CALCULATES THE DIMENSIONS AT A LATER
C                 TIME
C
C    CLDIM        DETERMINES THE OUTPUT VARIABLES DESCRIBING THE CLOUD
C                 DIMENSIONS.
C
C    TRNCAL       CONTROLING ROUTINE FO THE CALCULATION OF TRANMITTANCES.
C
C *******************************************************************
      IF(LOSTRN.OR.EDGE)GO TO 101
      NERR=4
      GO TO 999
  101 IF(.NOT.NEWATM) GO TO 200
      THETAX=THWND*PIRAD
      CALL ATMCAL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,THETAX,DHDT,ERR)
99999 IF(.NOT.ERR)GO TO 155
```

```
      NERR=7
      GO TO 999
  155 CONTINUE
C
C     COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT USER
C     DEFINED COORDINATES INTO LOCAL COORDINATES WITH X AXIS IN
C     THE WIND DIRECTION.
C
      THETAX=THWND*PIRAD
      TRNFRM(1,1)=COS(THETAX)
      TRNFRM(2,2)=TRNFRM(1,1)
      TRNFRM(1,2)=SIN(THETAX)
      TRNFRM(2,1)=-TRNFRM(1,2)
  200 CONTINUE
      IF(.NOT.NEWSRC) GO TO 300
      TWIND=1.E5
      TTR=1.E5
      TPRES=0.
      DEL=.001
      DO 250 I=1,2
      IF(NARY.GT.1)SRCBAS(I)=COOR(I,1)
      ORIG(I)=SRCBAS(I)
  250 CONTINUE
      CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD)
      CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)
  300 CONTINUE
      IF(.NOT.LOSTRN) GO TO 400
C
C     CONVERT TRNCOR AND RECCOR TO LOCAL COORDINATES WITH ORIGIN AT
C     SRCBAS AND X AXIS IN WIND DIRECTION.
C
      TRN(3)=TRNCOR(3)
      REC(3)=RECCOR(3)
      DO 320 I=1,2
      TRN(I)=0.
      REC(I)=0.
      DO 310 J=1,2
      TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))
      REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))
  310 CONTINUE
  320 CONTINUE
  400 CONTINUE
      IF(.NOT.EDGE) GO TO 500
C
C     COMPUTE A UNIT VECTOR IN THE DIRECTION OF THE OBSERVERS LINE
C     OF SIGHT AND A UNIT VECTOR PERPENDICULAR TO THE LINE OF SIGHT
C
      CALL VSUM(ORIG,OBSCOR,ONEM,DIR)
      CALL UNIT(DIR,DIR,RANGE)
      COSTH=0.
      SINTH=0.
      DO 410 J=1,2
      COSTH=COSTH+TRNFRM(1,J)*DIR(J)
      SINTH=SINTH+TRNFRM(2,J)*DIR(J)
  410 CONTINUE
      SINTH2=SINTH*SINTH
      COSTH2=COSTH**2
      SCRN(1)=SINTH
      SCRN(2)=-COSTH
C
C      COMPUTE THE PROJECTION OF EACH DIFFERENCE VECTOR DIFF ONTO THE
C      VECTOR PERPENDICULAR TO THE LINE OF SIGHT,(DIFF(1,J),DIFF(2,J))
C      IS THE VECTOR FROM THE REFERENCE CHARGE TO THE JTH CHARGE
C      LOCATION IN THE INTERNAL COORDINATE SYSTEM.
C
      PARY1=0.0
      PARY2=0.0
      DO 420 J=1,NCHTOT
      DO 415 I=1,2
      TEMP(I)=DIFF(I,J)
```

```
  415 CONTINUE
      PRSEP(J)=DOTPRD(TEMP,SCRN)
      X=PRSEP(J)
      IF(X.LT.0.0)GO TO 416
      IF(X.GT.PARY1)PARY1=X
      GO TO 420
  416 IF(X.LT.PARY2)PARY2=X
  420 CONTINUE
      PRJARY=(PARY1-PARY2)/2.
      CENDIF=(PARY1+PARY2)/2.
  500 CONTINUE
      IF(NARY.EQ.3)GO TO 600
      IF(NEWTIM) CALL RISE(TPRES,TIME,DEL)
  600 IF(.NOT.EDGE) GO TO 650
      FTIME=TIME
      CALL CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS,ERR)
      IF(.NOT.ERR)GO TO 650
      NERR=6
      GO TO 999
  650 CONTINUE
      IF(.NOT.LOSTRN)GO TO 999
C
      CALL TRNCAL(TRN,REC,TIME,TRNLOS)
  999 RETURN
      END
```

```
      FUNCTION ERF(X)
C     CALCULATES THE ERROR FUNCTION
C
C  INPUT
C
C     X     VALUE AT WHICH ERROR FUNCTION IS DESIRED
C
C  FUNCTIONS AND SUBROUTINES NEEDED
C
C     NONE
C
C*********************************************************************
      DIMENSION P(3,5),Q(3,5)
      DATA RPI/.5641896/
      DATA P/2.138533E+01,7.373888E+00,-4.257996E-02,
     +       1.722276E+00,6.865018E+00,-1.960690E-01,
     +       3.166529E-01,3.031799E+00,-5.168823E-02,
     +       0.          ,5.631696E-01,0.           ,
     +       0.          ,4.318779E-05,0.           /
      DATA Q/1.895226E+01,7.373961E+00,1.509421E-01,
     +       7.843746E+00,1.518491E+01,9.214524E-01,
     +       1.000000E+00,1.279553E+01,1.000000E+00,
     +       0.          ,5.354217E+00,0.           ,
     +       0.          ,1.000000E+00,0.           /
      AX=ABS(X)
      ERFC=0.0
      IF(AX.GT.11.0) GO TO 300
      X2=AX*AX
      I=2
      IF(AX.LT.0.5) I=1
      IF(AX.GT.4.) I=3
      IF(I-2) 10,20,30
   10 N=3
      Z=X2
      GO TO 40
   20 N=5
      Z=AX
      GO TO 40
   30 N=3
      Z=1./X2
   40 SP=P(I,N)
      SQ=Q(I,N)
      N1=N-1
      DO 50 K=1,N1
      J=N-K
      SP=SP*Z+P(I,J)
   50 SQ=SQ*Z+Q(I,J)
      IF(I-2) 60,70,80
   60 ERFC=1.0-X*SP/SQ
      ERF=1.-ERFC
      RETURN
   70 ERFC=EXP(-X2)*SP/SQ
      GO TO 300
   80 ERFC=EXP(-X2)/AX*(RPI+SP/(SQ*X2))
  300 IF(X.LT.0.0) ERFC=2.0-ERFC
      ERF=1.-ERFC
      RETURN
      END
```

```
      SUBROUTINE FIT(X,F,A,B,C)
C
C   QUADRATIC FIT TO THREE POINTS USING NEWTON'S FUNDAMENTAL FORMULA
C
C   INPUTS
C
C      X     - 3 VALUES OF THE INDEPENDENT VARIIABLE
C
C      F     - 3 FUNCTION VALUES CORRESPONDIING TO THE X VALUES
C
C   OUTPUTS
C
C      A     - COEFFICIENT OF THE  X**2  TERM
C
C      B     - COEFFICIENT OF THE  X  TERM
C
C      C     - CONSTANT TERM
C
C*****************************************************************
      DIMENSION X(3),F(3)
      H=X(2)-X(1)
      DF1=(F(2)-F(1))/H
      DF2=(F(3)-2.*F(2)+F(1))/(2.*H**2)
      A=DF2
      B=DF1-DF2*(X(2)+X(1))
      C=F(1)+X(1)*(X(2)*DF2-DF1)
      RETURN
      END
```

```
      FUNCTION FUNCT(X,Z)
      LOGICAL HORIZ,SKIP
      COMMON /CLOCK/ TIME,TWIND
      COMMON/MODE/HORIZ
      COMMON/SKIPIT/SKIP
C ***********************************************
C
C
C     PURPOSE
C
C       TO SUPPLY A TRANSMITTANCE FUNCTION FOR THE CONTOUR TRACING
C       ROUTINE IN ORDER TO DETERMINE THE CLOUD EDGE.
C
C     INPUT
C
C       X    THE HORIZONTAL COORDINATE IN METERS
C       Z    THE VERTICAL COORDINATE IN METERS
C
C     OUTPUT
C
C       RETURNS THE LOG OF THE OPTICALLY WEIGHTED CL VALUE (AT
C       VISIBLE WAVELENGTHS) FOR THE LINE OF SIGHT SPECIFIED BY X,Z
C
C     FUNCTIONS CALLED
C
C       CWIND
C
C     CALLED BY GFUN, CLIMB, GRAD2
C
C *********************************************************
      HORIZ=.TRUE.
      SKIP=.TRUE.
      Y=0.
      EXT1=0.0
      EXT2=0.0
      IF(Z.LE.0.)GO TO 100
      EXT1=CWIND(X,Y,Z,TIME)
      IF(TIME.LE.TWIND)GO TO 10
      EXT2=CSPHER(X,Y,Z,TIME)
   10 EXT=EXT1+EXT2
      IF(EXT.LE.1.E-30)GO TO 100
      FUNCT=ALOG(EXT)
      GO TO 999
  100 FUNCT=-30.
  999 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GAMMA(XX,GX,IER)
C
C        SUBROUTINE GAMMA
C
C        PURPOSE
C           COMPUTES THE GAMMA FUNCTION FOR A GIVEN ARGUMENT
C
C        USAGE
C           CALL GAMMA(XX,GX,IER)
C
C        DESCRIPTION OF PARAMETERS
C           XX  -THE ARGUMENT FOR THE GAMMA FUNCTION
C           GX  -THE RESULTANT GAMMA FUNCTION
C           IER -THE RESULTANT ERROR CODE WHERE
C                IER=0  NO ERROR
C                IER=1  XX IS WITHIN .000001 OF BEING A NEGATIVE INTEGER
C                IER=2  XX GT 57, OVERFLOW, GX SET TO 1.E32
C
C        COMMENTS
C             NONE
C
C        SUBROUTINES AND FUNCTIONS
C             NONE
C
C        METHOD
C           THE RECURSION RLATION AND POLYNOMIAL APPROXIMATION
C           BY C. HASTINGS,JR., 'APPROXIMATIONS FOR DIGITAL COMPUTERS',
C           PRINCETON UNIVERSITY PRESS, 1955
C
C     ..................................................................
C
      IF(XX-57.) 6,6,4
    4 IER=2
      GX=1.E32
      RETURN
    6 X=XX
      ERR=1.0E-6
      IER=0
       GX=1.0
      IF(X-2.0) 50,50,15
   10 IF(X-2.0) 110,110,15
   15 X=X-1.0
      GX=GX*X
      GO TO 10
   50 IF(X-1.0) 60,120,110
C
C        SEE IF X IS NEAR NEGATIVE INTEGER OR ZERO
C
   60 IF(X-ERR) 62,62,80
   62 Y=FLOAT(INT(X))-X
      IF(ABS(Y)-ERR) 130,130,70
C
C        X NOT NEAR A NEGATIVE INTEGER OR ZERO
C
   70 IF(X-1.0)80,80,110
   80 GX=GX/X
      X=X+1.0
      GO TO 70
  110 Y=X-1.0
      GY=1.0+Y*(-0.5771017+Y*(0.9858540+Y*(-0.8764218+Y*(0.8328212+
     1Y*(-0.5684729+Y*(0.2548205+Y*(-0.05149930)))))))
      GX=GX*GY
  120 RETURN
  130 IER=1
      RETURN
      END
```

```
      FUNCTION GFUN(S)
C
C       GFUN IS THE RESTRICTION OF THE TWO DIMENSIONAL FUNCTION, F, TO
C               A LINE. I.E. FORM G(S)=F(X,Y), WHERE (X,Y)=BASE+S*DIR.
C
      EXTERNAL FUNCT
      DIMENSION                P(2)
      COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
      CALL VSUM(BASE,DIR,S,P)
      GFUN=FUNCT(P(1),P(2))
      RETURN
      END
```

```
      SUBROUTINE GRAD2(PT,FUNCT,RES,GRAD,SLOPE)
C
C  THIS SUBROUTINE CALCULATES THE UNIT GRADIENT VECTOR
C  OF A FUNCTION AT THE GIVEN POINT USING THE FORMULA
C  PARTIAL DF/DX = (F(X+R,Y)-F(X,Y))/R WHERE R IS
C  SMALL, SIMILARLY FOR PARTIAL DF/DY. IT THEN
C  NORMALIZES THE RESULTANT VECTOR FOR THE UNIT GRADIENT
C  AND FINDS THE SLOPE,WHICH IS THE MAGNITUDE OF
C  OF THE REGULAR GRADIENT.
C
C  ARGUMENTS PASSED
C      PT-THE POINT AT WHICH THE UNIT GRADIENT IS FOUND.
C      FUNCT-THE FUNCTION(X,Y)
C      RES-R
C      GRAD-THE UNIT GRADIENT
C      SLOPE-SLOPE AT PT
C
C  OTHER VARIABLES
C      C00,C10,C01-THE FUNCTION AT THE POINTS F(X,Y),F(X+R,Y),
C          AND F(X,Y+R) RESPECTIVELY.
C
C  SUBROUTINES CALLED
C      UNIT-NORMALIZES A VECTOR AND FINDS ITS MAGNITUDE.
C
      DIMENSION PT(2),GRAD(2)
      C00=FUNCT(PT(1),PT(2))
      C10=FUNCT(PT(1)+RES,PT(2))
      C01=FUNCT(PT(1),PT(2)+RES)
      GRAD(1)=(C10-C00)/RES
      GRAD(2)=(C01-C00)/RES
      CALL UNIT(GRAD,GRAD,SLOPE)
      RETURN
      END
```

```
      SUBROUTINE GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)
C
C   ROUTINE TO EVALUTE THE INTEGRAND FOR THE TRAPEZOIDAL INTEGRATION
C   FOR FINDING THE OPTICALLY WEIGHTED CONCENTRATION ALONG THE LINE OF
C   SIGHT WHERE THE DUST IS GENERATED BY A VEHICLE.
C
C   INPUTS
C
C      U     - A UNIT VECTOR ALONG THE LINE OF SIGHT
C
C      TR    - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATE
C              SYSTEM
C
C     XNORM - DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER
C
C     TIME - PRESENT TIME AT WHICH A TRANSMITTANCE IS WANTED
C
C     TIVEN- TIME THAT THE VEHICLE HAS TRAVELED
C
C     VDIR - VECTOR CONTAINING DESCRIBING THE VEHICLE DIRECTION AND
C            SPEED
C
C   OUPUT
C
C     VALUE - VALUE OF THE INTEGRAND
C
C   FUNCTIONS AND SUBROUTINES NEEDED
C
C      CONLEN - TO FIND THE LENGTH OF THE INTERSECTION OF THE LINE OF
C               SIGHT AND THE TILTED CYLINDER
C
C*********************************************************************
      DIMENSION U(3),TR(3),VDIR(2),VP(2)
      COMMON/MO5/DMMMY(604),DMM(600),
     +              ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
     +              RB(3,25),Z2(3,25)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/VL/VLOAD
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C
C   FIND THE VEHICLE POSITION AT TIME TIVEH
C
      VP(1)=TIVEH*VDIR(1)
      VP(2)=TIVEH*VDIR(2)
      TOF=TIME-TIVEH
      DO 10 I=1,ICOUNT
      IND=I
      IF(TOF.LT.TIMES(I))GO TO 20
   10 CONTINUE
C
C   IF TOF (TIME OF FLIGHT) IS GREATER THAN TABULATED VALUES IT IS
C   ASSUMED THE THE CLOUD HAS DISSIPATED
C   XC=X1*Z+X0  IS THE LINE THRIUGH THE CENTER OF THE CYLINDER
C
      GO TO 50
   20 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)
      X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)
      RAD=TOF*(RT(1,IND)*TOF+RT(2,IND))+RT(3,IND)
      HTTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)
      HTBOT=0.0
      RTOP=RAD
      RBOT=RAD
      XCEN=VP(1)+(X1*HTTOP+X0)
      YCEN=VP(2)
      XB=VP(1)+(X1*HTBOT+X0)
      YB=VP(2)
      IF(U(3).LT.1.E-06)GO TO 30
C
C   COMPUTE INTERSECTION LENGTH FOR NON HORIZONTAL LINES OF SIGHT
C
```

```
      CALL CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,XNORM,PLEN)
      GO TO 40
C
C  DETERMINE LENGTH OF INTERSECTION FOR A HORIZONTAL LINE OF SIGHT
C
   30 IF(HTTOP.LT.TR(3))GO TO 35
      A=U(1)**2+U(2)**2
      B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)
      C=(TR(1)-XCEN)**2+(TR(2)-YCEN)**2-RAD**2
      X=B**2-A*C
      IF(X.LT.0.0)GO TO 35
      P1=(-B+SQRT(X))/A
      P2=(-B-SQRT(X))/A
      IF(P1.GT.XNORM.AND.P2.GT.XNORM)GO TO 35
      IF(P2.LT.0.0.AND.P1.LT.0.0)GO TO 35
      PLEN=AMIN1(P1,XNORM)-AMAX1(P2,0.0)
      GO TO 40
   35 PLEN=0.0
   40 VOL=PI*HTTOP*(RAD**2)
      VALUE=VLOAD*PLEN/VOL
      GO TO 999
   50 VALUE=0.0
  999 RETURN
      END
```

```
      SUBROUTINE GREEN(Z,Z1,T,ALPHA,T0,IER)
C ***********************************************************
C
C
C     PURPOSE
C
C       TO COMPUTE THE GENERALIZED GREENS FUNCTION
C
C      USES GREEN1
C
C      SEE GREEN1 FOR ARGUMENT LIST
C
C
C ************************************************************
      REAL N,M
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
C
      IF(N .EQ. 1.) GO TO 2
      X2=2.-N
      AT=ALPHA*T
      T0=0.
      IF(AT .GE. Z1) RETURN
      CALL GREEN1((Z+AT)**X2,Z1**X2,X2*X2*T,(N-1.)/X2,T1,IER)
      T1=T1*X2*Z1**(1.-N)
      U=1.
      T2=0.
      IF(ABS(ALPHA) .LT. 1.E-4) GO TO 1
      ZMZ=Z-Z1+AT
      X2=N+1.
      AN1=ALPHA*X2
      ZMZN=Z1**X2 - (Z1-AT)**X2
      ARG=(-AN1*ZMZ*ZMZ)/(4.*ZMZN)
      IF(ARG .LT. -70.) GO TO 3
      T2=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
    3 IF(T1.LT.1.E-30 .AND. T2.LT.1.E-30) RETURN
C
C     CALCULATION OF MIXING RATIO, U, BY N=1 ANALOGY
C
      CALL GREEN1(Z+AT,Z1,T,0.,T1U,IER)
      X2=2.
      AN1=ALPHA*X2
      ZMZN=Z1**X2 - (Z1-AT)**X2
      T2U=0.
      ARG=(-AN1*ZMZ*ZMZ)/(4.*ZMZN)
      IF(ARG .LT. -70.) GO TO 4
      T2U=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
    4 IF(T1U.LT.1.E-30 .AND. T2U.LT.1.E-30) GO TO 1
      CALL GREEN1(Z,Z1,T,ALPHA,G,IER)
      U=(G-T2U)/(T1U-T2U)
    1 IF(U .LT. 0.) U=0.
      IF(U .GT. 1.) U=1.
C
C     COMBINE LIMITING SOLUTIONS WITH DETERMINED MIXING RATIO
C
      T0=U*T1 + (1.-U)*T2
      RETURN
    2 CALL GREEN1(Z,Z1,T,ALPHA,T0,IER)
      RETURN
      END
```

```
      SUBROUTINE GREEN1(Z,Z1,T,NU,BI,IER)
C
C        SUBROUTINE GREEN1
C
C        PURPOSE
C           COMPUTE THE I BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
C                 AND MULTIPLY BY AN APPROPRIATE POWER OF THE ARGUMENT
C                 AND AN EXPONENTIAL IN ORDER TO CALCULATE THE GREENS
C                 FUNCTION FOR THE WIND DIFFUSION EQUATION
C
C        USAGE
C           CALL GREEN1(Z,Z1,T,NU,BI,IER)
C
C        DESCRIPTION OF PARAMETERS
C           Z,Z1,T    -THE ARGUMENTS OF THE FUNCTION DESIRED
C           NU  -THE ORDER OF THE I BESSEL FUNCTION
C           BI  -THE RESULTANT BESSEL FUNCTION
C           IER -RESULTANT ERROR CODE WHERE
C              IER=-1  EXPONENTIAL UNDERFLOW (NON-FATAL),  BI SET TO 0.0
C              IER=0 NO ERROR
C              IER=1 NU NEAR NEGATIVE INTEGER
C              IER=2 OVERFLOW IN GAMMA
C              IER=3 UNDERFLOW, BI .LT. 1.E-32, BI SET TO 0.0
C              IER=4 OVERFLOW, X .GT. 90 WHERE X .GT. N
C              IER=5 X IS NEGATIVE
C
C        REMARKS
C           NU IS A  REAL NUMBER
C           N AND X MUST BE .GE. ZERO
C           THIS SUBROUTINE IS A MODIFICATION OF BESI WHICH COMPUTES THE
C           I BESSEL FUNCTION FOR INTEGER ORDERS.  THE CHANGE REQUIRES
C           USE OF THE GAMMA FUNCTION FOR COMPUTING THE FIRST TERM OF THE
C           SERIES.  THE SUCCESSIVE TERMS ARE CALCULATED WITH THE SAME
C           RECURSION FORMULA AND THE ASYMPTOTIC APPROXIMATION IS ALSO
C           UNCHANGED.  BESI IS IN THE IBM SYSTEM/360 SCIENTIFIC
C           SUBROUTINE PACKAGE.  MODIFICATIONS MADE BY D. DVORE, AERODYNE
C           RESEARCH  INC. JANUARY 15,1979.
C
C        SUBROUTINES AND FUNCTIONS REQUIRED
C           GAMMA  WHICH COMPUTES THE GAMMA FUNCTION
C
C        METHOD
C           COMPUTES I BESSEL FUNCTION USING SERIES OR ASYMPTOTIC
C           APPROXIMATION DEPENDING ON THE RANGE OF THE ARGUMENT.
C
C     CALLED BY MOMENT
C
C     .............................................................
C
      REAL NU
      X=2.*SQRT(Z*Z1)/T
C
C     CHECK FOR ERRORS IN NU AND X AND EXIT IF ANY ARE PRESENT
C
      IER=0
      BI=1.0
      IF(NU)10,15,10
   10 IF(X)160,20,20
   15 IF(X)160,17,20
   17 ARG=-(Z+Z1)/T
      IF(ARG .LT. -80.) GO TO 170
      BI=EXP(ARG)/T
      RETURN
C
C     DEFINE TOLERANCE
C
   20 TOL=1 E-3
C
C     IF ARGUMENT GT 12 AND GT NU, USE ASYMPTOTIC FORM
C
```

```
      IF(X-12.)40,40,30
   30 IF(X-ABS(NU))40,40,110
C
C     COMPUTE FIRST TERM OF SERIES AND SET INITIAL VALUE OF THE SUM
C
   40 XX=X/2.
      N=INT(NU)
      FN=N
      R=NU-FN
      CALL GAMMA(1.+NU,GR,IER)
      IF(IER .EQ. 0) GO TO 60
   50 BI=0.0
      RETURN
   60 TERM=1./GR
   70 BI=TERM
      XX=XX*XX
C
C     COMPUTE TERMS, STOPPING WHEN ABS(TERM) LE ABS(SUM OF TERMS)*TOLERA
C
      DO 90 K=1,1000
      IF(ABS(TERM)-ABS(BI*TOL))95,95,80
   80 FK=K
      FK=FK*(NU+FK)
      TERM=TERM*(XX/FK)
   90 BI=BI+TERM
   95 ARG=-(Z+Z1)/T
      IF(ARG .LT. -80.) GO TO 170
      BI=BI*(Z1/T)**NU*EXP(ARG)/T
C
C     RETURN BI AS ANSWER
C
  100 RETURN
C
C     X GT 12 AND X GT NU, SO USE ASYMPTOTIC APPROXIMATION
C
  110 FN=4.*NU*NU
  115 XX=1./(8.*X)
      TERM=1.
      BI=1.
      DO 130 K=1,30
      IF(ABS(TERM)-ABS(BI*TOL)) 140,140,120
  120 FK=(2*K-1)**2
      TERM=TERM*XX*(FK-FN)/FLOAT(K)
  130 BI=BI+TERM
C
C     SIGNIFICANCE LOST AFTER 30 TERMS, TRY SERIES
C
      GO TO 40
  140 PI=3.141592653
      ARG=X-(Z+Z1)/T
      IF(ARG .LT. -80.) GO TO 170
      BI=BI*(Z1/Z)**(NU/2.)*EXP(ARG)/SQRT(2.*PI*X)/T
      GO TO 100
  160 IER=5
      GO TO 100
  170 BI=0.0
      GO TO 50
      END
```

```
      SUBROUTINE MOMENT(VGRAV,ZIN,H,TIN,Q,XBAR,SIGW2,SIGP2)
      REAL M,N,NM
      DIMENSION AL(9),Z(9),T(9),XB(81,4,3),SW(81,4,3),SP(81,4,3),NM(9)
      DIMENSION VAL(16),XVAL(8),W(8),XI(4),IB(4),NTC(4),II(4),X(9,4)
      LOGICAL FIRST
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      EQUIVALENCE (Z(1),X(1,1)),(T(1),X(1,2)),(AL(1),X(1,3))
      EQUIVALENCE (NM(1),X(1,4))
      DATA FIRST/.TRUE./,IB/4,4,1,2/,ITC/11/,HREF/1./
C ***************************************************************
C
C
C     PURPOSE
C
C       TO CONVERT PARAMETERS TO NONDIMENSIONAL FORM AND THEN COMPUTE
C     THE ZERO ORDER MOMENT AND INTERPOLATE FROM TABULATED VALUES OF
C     THE HIGHER ORDER MOMENTS
C
C     INPUT
C
C       VGRAV   THE GRAVITATIONAL SETTLING VELOCITIES OF THE PARTICLE
C               IN METERS / SEC
C       ZIN       THE HEIGHT (METERS) AT WHICH THE MOMENTS ARE DESIRED
C       H       THE HEIGHT OF RELEASE OF THE PARTICLES IN METERS
C       TIN     THE TIME IN SECONDS AFTER RELEASE
C
C     OUTPUT
C
C       Q       THE VERTICAL CONCENTRATION OF PARTICLES AT HEIGHT Z
C       XBAR    THE DISPLACEMENT (METERS) IN THE X (IE WIND) DIRECTION
C               OF THE CENTER OF MASS OF PARTICLES AT HEIGHT Z
C       SIGW2   THE SQUARE OF THE STANDARD DEVIATION OF THE WINDWARD
C               DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2
C       SIGP2   THE SQUARE OF THE STANDARD DEVIATION OF THE CROSS-WIND
C               DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2
C
C
C     SUBROUTINES CALLED
C
C       DTERPS    PUTS THE NEEDED VALUES OF THE TABULATED MOMENTS
C                 INTO A ONE-DIMENSIONAL ARRAY
C       DTERPI    A FUNCTION WHICH RETURNS THE INTERPOLATED VALUE
C                 FOR GIVEN ARGUMENTS AND ARRAYS
C       GREEN     CALCULATES THE GREENS FUNCTION WHICH IS THE
C                 0-ORDER MOMENT
C
C     CALLED BY CWIND
C
C ***************************************************************
      IF(.NOT.FIRST)GO TO 5
C
C     READ IN THE TABLE OF MOMENTS ON THE FIRST CALL OF MOMENT
C
C      Z      LOG OF NON-DIMENSIONAL HEIGHTS AT WHICH MOMENTS ARE TABULA
C      T      LOG OF NON-DIMENSIONAL TIMES   AT WHICH MOMENTS ARE TABULA
C      AL     NON-DIMENSIONAL SETTLING VELOCITIES AT WHICH MOMENTS ARE
C                   TABULATED
C      NM     DIFFUSIVITY POWER LAW EXPONENTS AT WHICH MOMENTS ARE
C                     TABULATED
C
C      XB     TABULATED VALUES OF LOGS OF FIRST ORDER MOMENTS (RELATED
C                  TO MEAN HORIZONTAL DISPLACEMENT)
C      SW     TABULATED VALUES OF LOGS OF WIND SHEAR COMPONENT OF SECOND
C              ORDER MOMENT (CONTRIBUTES TO VARIANCE IN WIND DIRECTION)
C      SP     TABULATED VALUES OF LOGS OF SECOND ORDER MOMENT COMMON TO
C                 WIND AND CROSS-WIND VARIANCES
C
      READ(NDIRTU,1) NZ,NT,NA,NN
    1 FORMAT(4I3)
```

```
      NTC(1)=NZ-1
      NTC(2)=NT-1
      NTC(3)=NA-1
      NTC(4)=NN-1
      READ(NDIRTU,2) (Z(I),I=1,NZ)
    2 FORMAT(6E13.5)
      READ(NDIRTU,2) (T(I),I=1,NT)
      READ(NDIRTU,2) (AL(I),I=1,NA)
      READ(NDIRTU,2) (NM(I),I=1,NN)
      NZT=NZ*NT
      DO 3 L=1,NN
      READ(NDIRTU,2) ((XB(IJ,K,L),IJ=1,NZT),K=1,NA)
      READ(NDIRTU,2) ((SW(IJ,K,L),IJ=1,NZT),K=1,NA)
      READ(NDIRTU,2) ((SP(IJ,K,L),IJ=1,NZT),K=1,NA)
    3 CONTINUE
      FIRST=.FALSE.
      REWIND NDIRTU
    5 CONTINUE
C
C     CONVERT INPUT PARAMETERS TO NONDIMENSIONAL FORM
C
      SCLU=DZ0*H**(N-1.)
      XI(1)=ZIN/H
      XI(2)=SCLU*TIN/H
      XI(3)=VGRAV/SCLU
      XI(4)=N
      CALL GREEN(XI(1),HREF,XI(2),XI(3),Q,IER)
      Q=Q/H
      IF(Q .LE. 1.E-10) GO TO 999
C
C     TAKE LOGS FOR LOGARITHMIC INTERPOLATION
C
      XI(1)=ALOG(XI(1))
      XI(2)=ALOG(XI(2))
C
C     DETERMINE INDICES OF LOWEST CORNER POINT OF THE CUBE TO
C     BE USED IN INTERPOLATION MAKING SURE THAT ENOUGH CORNER POINTS
C     OF THE CUBE HAVE TABULATED VALUES
C
      DO 100 I=1,4
      II(I)=IB(I)
100   CONTINUE
      DO 101 III=1,4
      I=5-III
6     IA=II(I)
      IF(XI(I) .GE. X(IA,I) .AND. XI(I) .LE. X(IA+1,I)) GO TO 101
      IF(XI(I) .LT. X(IA,I) .AND. IA .EQ. 1) GO TO 101
      IF(XI(I) .GT. X(IA,I) .AND. IA .EQ. NTC(I)) GO TO 101
      ISAV=II(I)
      II(I)=IA + IFIX(SIGN(1.,XI(I)-X(IA,I)))
      IT=0
      DO 102 JI=1,2
      JIX=JI + II(1) - 1
      DO 102 IJ=1,2
      IJX=JIX + (IJ + II(2) - 2)*NZ
      DO 102 K=1,2
      KX=K-1 + II(3)
      DO 102 L=1,2
      LX=L-1 + II(4)
      IF(XB(IJX,KX,LX) .GT. -100.) IT=IT+1
102   CONTINUE
      IF(IT .GT. ITC) GO TO 6
      II(I)=ISAV
101   CONTINUE
C
C     PERFORM THE INTERPOLATION WITH DETERMINED CUBE OF POINTS
C
      DO 103 I=1,4
      I2=I*2
      I1=I2-1
```

```
      IA=II(I)
      XVAL(I1)=X(IA,I)
      XVAL(I2)=X(IA+1,I)
103   CONTINUE
      CALL DTERPS(II,XB,VAL,NZ)
      XBAR=DTERPI(4,XI,XVAL,VAL,-100.,W)
      CALL DTERPS(II,SW,VAL,NZ)
      SIGW2=DTERPI(-4,XI,XVAL,VAL,-100.,W)
      CALL DTERPS(II,SP,VAL,NZ)
      SIGP2=DTERPI(-4,XI,XVAL,VAL,-100.,W)
C
C     CONVERT THE LOG OF THE NONDIMENSIONAL VALUES INTERPOLATED
C     TO THE USUAL DIMENSIONAL FORM
C
      SCL=U0*H**(M+1.)/SCLU
      XBAR=SCL*EXP(XBAR)
      SIGW2=SCL*SCL*EXP(SIGW2)
      SIGP2=2.*DXZ0*H*H*EXP(SIGP2)
      SIGW2=SIGW2+SIGP2
      SIGP2=DYX0*SIGP2
999   RETURN
      END
```

```
      SUBROUTINE PATH(T,U,XCEN,YCEN,RAD,PLEN)
C
C    THIS SUBROUTINE COMPUTES THE PATH LENGTH THROUGH THE SPHERE OR
C    WAKE FOR A HORIZONTAL PATH.
C
C    INPUTS
C
C       T       - TRANSMITTER COORDINATE IN THE LOCAL COORDINATE SYSTEM
C
C       U       - UNIT VECTOR ALONG THE LINE CONNECTING THE TRANSMIITTER
C                 AND RECEIVER
C
C       XCEN    - X COORDINATE OF THE CENTER OF THE CIRCLE
C
C       YCEN    - Y COORDINATE OF THE CENTER OF THE CIRCLE
C
C       RAD     - RADIUS AT THE DESIRED HEIGHT
C
C    OUTPUT
C
C       PLEN    - LENGTH OF THE INTERSECTION OF THE CONE AT HEIGHT T(3)
C                 AND THE LINE OF SIIGHT
C
C
C    FUNCTIONS AND SUBROUTINES NEEDED
C
C       NONE
C
C*********************************************************************
      DIMENSION T(3),U(3)
      A=U(1)**2+U(2)**2
      PLEN=0.
      X=RAD**2*A-(U(2)*(T(1)-XCEN)-U(1)*(T(2)-YCEN))**2
      IF (X.GT.0.) PLEN=2.*SQRT(X)/A
      RETURN
      END
```

```
      SUBROUTINE PERP(A,B)
      DIMENSION A(2),B(2)
C *** B IS ROTATED 90 DEGREES COUNTERCLOCKWISE FROM A
      B(1)=-A(2)
      B(2)=A(1)
      RETURN
      END
```

```
      SUBROUTINE PRECL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,PHI,DHDT,
     1               CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT)
C
C     ROUTINE TO PRECOMPUTE EXPLOSION PRODUCED DUST CLOUD AND STORE ON
C     EXTERNAL FILE UNIT IFILE
C
C     OUTPUTS
C
C         XC1 COEFFICIENTS OF QUADRATIC FIT TO SLOPE OF LINE DESCRIBING
C             X DISPLACEMENT OF CONE (ALONGWIND)
C
C         XC0 COEFFICIENTS OF QUADRATIC FIT TO CONSTANT TERM OF LINE
C             DESCRIBING X DISPLACEMENT OF CONE (ALONGWIND)
C
C         Z2  COEFFICIENTS OF QUADRATIC FIT TO HEIGHT OF TOP OF CLOUD
C
C         RT  COEFFICIENTS OF QUADRATIC FIT TO THE RADIUS OF THE TOP OF THE
C             CONE
C
C         RB  COEFFICIENTS OF THE QUADRATIC FIT TO THE RADIUS OF THE CONE
C             AT A HEIGHT OF THE AVERAGE OF THE DISC SOURCES
C
C
C     THE ABOVE OUTPUT ARE THE COEFFICIENTS OF QUADRATIC FITS THROUGH
C     THREE CONSECUTIVE POINT IN TIME. THE QUADRATIC FITS ARE STORED IN
C     COMMON/  MO5 / WITH THE ARRAY TIMES CONTAINING THE LAST TIME
C     OF EACH QUADRATIC PIECE WITH THE FIRST STARTING AT 0.0
C     THE FITS ARE WRITTEN ONTO A FILE INDICATED BY IFILE USING
C     A BINARY WRITE
C     THE FITS ARE STORED SUCH THAT
C
C         F(TIME)=VAR(1,J)*TIME**2 + VAR(2,J)*TIME + VAR(3,J)
C
C           AND       TIMES(J-1) < TIME < TIMES(J)
C
C*****************************************************************
      LOGICAL SWITCH,CHANGE,DHDT
      REAL KZ,KX
      DIMENSION T(3),FRB(3),FRT(3),FXC1(3),FXC0(3),XB(3),OWF(5,2)
      DIMENSION FZ2(3),ZTEMP(2),TMPMES(2),ZWND(2),WNDMES(2),OWFC(5)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/OPTION/IOPT,IFILE
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
      COMMON/BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),TIM
      COMMON/MO5/DMMMY(604),DMM(600),
     +            ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
     +            RB(3,25),Z2(3,25)
      COMMON/VL/VLOAD
      COMMON/SIG/SIG02,SIGC
      COMMON/CLOCK/FTIME,TWIND
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON/CARB/RCARB1,RCARB2
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
      DATA OWFC/1.,.95,.5,.2,1.E-03/
      TINC=1.0
      TPRES=0.0
      ICOUNT=0
      DEL=.001
      T(3)=0.0
      SUM2=0.0
C
C     FIND AVERAGE OF THE DISC RELEASE HEIGHTS AND THE AVERAGE OF THE
C     INITIAL SPREADS OF THE DISCS
C
      DO 5 J=1,NDSCS
      SUM2=SUM2+R2DSC(J)
    5 CONTINUE
```

```
      ZREF=2.0
      ROH2=SUM2/NDSCS
   10 ICOUNT=ICOUNT+1
      DO 50 I=1,3
      T(I)=T(3)+FLOAT(I-1)*TINC
C
C  FIND THE AVERAGE OF THE MOMENTS AT HEIGHT ZREF
C
      CALL AVRG(ZREF,T(I),QTOT,XBAVRG,SIG2X,SIG2Y)
      IF(QTOT.LT.1.E-10)GO TO 15
      SIGX=SQRT(SIG2X+ROH2/2.)
      SIGY=SQRT(SIG2Y+ROH2/2.)
      FRB(I)=SQRT(SIGX*SIGY)*1.5
      XB(1)=XBAVRG
      GO TO 20
   15 CALL WIN(ZREF,UX,V)
      XB(1)=UX*T(I)
      FRB(I)=0.0
   20 IF(T(I).GT.TWIND)GO TO 30
      CALL RISE(TPRES,T(I),DEL)
      FZ2(I)=ZCM+(2./3.)*RSPH
      IF(FZ2(I).GT.ZINV)FZ2(I)=ZINV
      FRT(I)=RSPH
      XB(2)=XCM
      GO TO 40
   30 XB(2)=XTR+VTR*(T(I)-TTR)
      SIGX2=SIG02+2.*KX*(T(I)-TTR)
      SIGZ2=SIG02+2.*KZ*(T(I)-TTR)
      SIGX=SQRT(SIGX2)
      SIGZ=SQRT(SIGZ2)
      SIG=SQRT(SIGX*SIGZ)
      FZ2(I)=ZTR+SIG
      FRT(I)=1.5*SIG
   40 FXC1(I)=(XB(2)-XB(1))/(FZ2(I)-ZREF)
      FXC0(I)=XB(1)-FXC1(I)*ZREF
   50 CONTINUE
C
C  COMPUT AND STORE QUADRATIC FITS
C
      TIMES(ICOUNT)=T(3)
C
C  FIT AND STORE RADIUS AT TOP
C
      CALL FIT(T,FRT,A,B,C)
      RT(1,ICOUNT)=A
      RT(2,ICOUNT)=B
      RT(3,ICOUNT)=C
C
C  FIT AND STORE RADIUS AT BOTTOM
C
      CALL FIT(T,FRB,A,B,C)
      RB(1,ICOUNT)=A
      RB(2,ICOUNT)=B
      RB(3,ICOUNT)=C
C
C  FIT AND STORE HEIGHT OF CLOUD
C
      CALL FIT(T,FZ2,A,B,C)
      Z2(1,ICOUNT)=A
      Z2(2,ICOUNT)=B
      Z2(3,ICOUNT)=C
C
C  FIT AND STORE XC1
C
      CALL FIT(T,FXC1,A,B,C)
      XC1(1,ICOUNT)=A
      XC1(2,ICOUNT)=B
      XC1(3,ICOUNT)=C
C
C  FIT AND STORE XC0
```

```
C
      CALL FIT(T,FXC0,A,B,C)
      XC0(1,ICOUNT)=A
      XC0(2,ICOUNT)=B
      XC0(3,ICOUNT)=C
C
C  CHECK TO SEE IF PRECOMPUTE CAN BE STOPPED
C
      IF(FRT(3).GE.FRB(3))PLEN=2.*FRT(3)
      IF(FRB(3).GE.FRT(3))PLEN=2.*FRB(3)
      VOL=(PI/3.)*(FZ2(3)-ZREF)*(FRT(3)**2+FRT(3)*FRB(3)+FRB(3)**2)
      NWL=1
      ACL=(RCARB1*DWF(NWL,NSOIL)+RCARB2*DWFC(NWL))*VLOAD*PLEN/VOL
      TINC=1.2*TINC
      IF(ICOUNT.LT.25.AND.ACL.GT..001)GO TO 10
C
C  WRITE PRECOMPUTED CLOUD INFORMATION ONTO FORTRAN UNIT IFILE WITH
C  A BINARY WRITE
C
      WRITE(IFILE)NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)
      WRITE(IFILE)DHDT,PHI,CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT,ZINV
      WRITE(IFILE)VLOAD,RCARB1,RCARB2
      WRITE(IFILE)ICOUNT
      DO 1000 J=1,ICOUNT
      WRITE(IFILE) TIMES(J),(RT(I,J),RB(I,J),Z2(I,J),XC0(I,J),
     1             XC1(I,J),I=1,3)
 1000 CONTINUE
      REWIND IFILE
  999 RETURN
      END
```

```
      SUBROUTINE PRETRN(TRN,REC,TIME,TRNLOS)
C
C  COMPUTE THE TRANSMITTANCE FOR THE RANDOM IN SPACE AND TIME
C  DISTRIBUTION OF CHARGES
C
C  INPUTS
C
C      TRN   -TRANSMITTER COORDINATES IN LOCAL COORDINATE SYSTEM
C
C      REC   -RECEIVER COORDINATES IN LOCAL COODINATE SYSTEM
C
C      TIME  -TIME AT WHICH TRANSMITTANCE IS DESIRED
C
C      NWL   -INTEGER INDEX FOR WAVELENGTH
C
C      NSOIL -SOIL TYPE
C
C  OUTPUTS
C
C      TRNLOS -TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT
C
C*****************************************************************
      LOGICAL TEST
      DIMENSION TRN(3),REC(3),TR(3),RE(3),OWF(5,2),U(3),OWFC(5)
      COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +            COOR(2,200),TSTAG(200),
     +               ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
     1               RB(3,25),Z2(3,25)
      COMMON/VL/VLOAD
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
      COMMON/CARB/RCARB1,RCARB2
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
      DATA OWFC/1.,.95,.5,.2,1.E-03/
C
C  PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER
C
      TEST=.FALSE.
      XNORM=0.0
      DO 10 I=1,3
      RE(I)=REC(I)
      TR(I)=TRN(I)
      U(I)=RE(I)-TR(I)
      XNORM=XNORM+U(I)**2
   10 CONTINUE
      XNORM=SQRT(XNORM)
      U(1)=U(1)/XNORM
      U(2)=U(2)/XNORM
      U(3)=U(3)/XNORM
C
C  COMPUTE THE CONTRIBUTION FROM EACH CHARGE TO THE OPTICALLY
C  WEIGHTED CONCENTRATION ALONG THE LINE OF SIGHT
C
      SUM=0.0
      DO 100 I=1,ITOT
      IF(TIME.LT.TSTAG(I))GO TO 100
      TOF=TIME-TSTAG(I)
      IF(TOF.GT.TIMES(ICOUNT))GO TO 100
      DO 20 J=1,ICOUNT
      IND=J
      IF(TOF.LE.TIMES(J))GO TO 30
   20 CONTINUE
C
C  DETERMINE NECESSARY PARAMETERS DESCRIBING THE CONICAL SHAPE SO THAT
C  THE LENGTH OF INTERSECTION OF THE LINE OF SIGHT AND CONE CAN BE
C  DETERMINED
C
   30 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)
      X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)
      HTTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)
```

```
      RTOP=TOF*(RT(1,IND)*TOF+RT(2,IND))+RT(3,IND)
      HTBOT=0.0
      RBOT=TOF*(RB(1,IND)*TOF+RB(2,IND))+RB(3,IND)
      XCEN=DIFF(1,I)+(X1*HTTOP+X0)
      YCEN=DIFF(2,I)
      XB=DIFF(1,I)+(X1*HTBOT+X0)
      YB=DIFF(2,I)
      IF(ABS(U(3)).LT.1.E-06)GO TO 40
C
C  COMPUTE THE INTERSECTION LENGTH FOR A NON-HORIZONTAL LINE OF SIGHT
C
      CALL CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,
     1            XNORM,PLEN)
      GO TO 50
C
C  COMPUTE THE INTERSECTION LENGTH FOR A HORIZONTAL LINE OF SIGHT
C
   40 IF(HTTOP.LT.TR(3))GO TO 45
      A=U(1)**2+U(2)**2
      B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)
C
C  DETERMINE THE RADIUS,X,AND Y POSITIONS OF THE CONE AT THE
C  TRANSMITTER HEIGHT
C
      ZETA=TR(3)/HTTOP
      RAD=ZETA*RTOP+(1.-ZETA)*RBOT
      XCEN=ZETA*XCEN+(1.-ZETA)*XB
      YCEN=ZETA*YCEN+(1.-ZETA)*YB
      A=U(1)**2+U(2)**2
      B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)
      C=(TR(1)-XCEN)**2+(TR(2)-YCEN)**2-RAD**2
      X=B**2-A*C
      IF(X.LT.0.0)GO TO 45
      P1=(-B+SQRT(X))/A
      P2=(-B-SQRT(X))/A
      IF(P1.GT.XNORM.AND.P2.GT.XNORM)GO TO 45
      IF(P2.LT.0.0.AND.P1.LT.0.0)GO TO 45
      PLEN=AMIN1(P1,XNORM)-AMAX1(P2,0.0)
      GO TO 50
   45 PLEN=0.0
   50 VOL=(PI/3.)*(HTTOP-HTBOT)*(RTOP**2+RTOP*RBOT+RBOT**2)
      CONT=VLOAD*PLEN/VOL
      SUM=SUM+CONT
      ACLSKT=0.0
      ACLSPH=0.0
      CALL TRNCHK(ACLSKT,SUM,ACLSPH)
      IF(TEST)GO TO 998
  100 CONTINUE
      TRNLOS=EXP(-SUM*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL)))
      GO TO 999
  998 TRNLOS=0.0
  999 RETURN
      END
```

```
      SUBROUTINE PREVEH(NSOIL,NWL)
C
C    ROUTINE FOR PRECOMPUTING VEHICLE GENERATED DUST CLOUD
C
C
C   INPUTS
C
C      NSOIL  - SOIL TYPE  (SEE DRTRAN FOR DETAILS)
C
C      NWL    - WAVELENGTH INDEX  (SEE DRTRAN)
C
C   OUTPUTS
C
C      RT     - COEFFICIENTS OF QUADRATIC FIT TO RADIUS OF THE CLOUD
C
C      Z2     - COEFFICIENTS OF QUADRATIC FIT TO HEIGHT OF THE CLOUD
C
C      XC1    - COEFFICIENTS OF QUADRATIC FIT TO SLOPE OF LINE DESCRIBING
C               X DISPLACEMENT OF CONE (ALONGWIND)
C
C      XC0    - COEFFICIENTS OF QUADRATIC FIT TO CONSTANT TERM OF LINE
C               DESCRIBING X DISPLACEMENT OF THE CONE  (ALONGWIND)
C
C   THE ABOVE OUTPUT ARE THE COEFFICIENTS OF QUADRATIC FITS THROUGH
C   THREE CONSECUTIVE POINTS IN TIME.  THE QUADRATIC FITS ARE STORED IN
C   COMMON/  M05 / WITH THE ARRAY TIMES CONTAINING THE LAST TIME OF THAT
C   INTERVAL.  THE FITS ARE WRITTEN ONTO A FILE INDICATED BY IFILE USING
C   A BINARY WRITE.
C
C   THE FITS ARE STORED SUCH THAT
C
C           F(TIME)=VAR(1,J)*TIME**2 + VAR(2,J)*TIME + VAR(3,J)
C
C      AND        TIMES(J-1) < TIME < TIMES(J)
C
C*********************************************************************
      LOGICAL HORIZ
      DIMENSION                        OWF(5,2),T(3),FR(3),FZ2(3)
      DIMENSION FXC1(3),FXC0(3),XB(2)
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
      COMMON/PRE/ZT0,RT2DZ
      COMMON/MODE/HORIZ
      COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,DMM(600),
     +              ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
     +              RB(3,25),Z2(3,25)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/PRTINF/R0,VGRAV(3),NPRTS
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
C
C   SET UP CORRECT PARAMETERS SO THAT CWIND CAN BE USED
C
      NARY=1
      ITOT=1
      NTOT=1
      NCHTOT=1
      HORIZ=.TRUE.
      ONEM=-1.
      COSTH=1.0
      SINTH=0.0
      COSTH2=COSTH**2
      SINTH2=SINTH**2
      SCRN(1)=SINTH
      SCRN(2)=-COSTH
      X=0.0
      Y=0.0
      TMAX=1000.
C
```

```
C     COMPUTE R , Z2 , XC1 , XC0 AT THREE CONSECUTIVE TIMES WITH SPACING
C     TINC AND THEN CALL FIT , WHICH CALCULATES A QUADRATIC FIT TO THESE
C     POINTS AND STORE THEM IN COMMON /QUADFT/
C
      TINC=1.0
      ICOUNT=0
      T(3)=0.0
   10 ICOUNT=ICOUNT+1
      DO 20 I=1,3
      T(I)=T(3)+FLOAT(I-1)*TINC
      ZREF=ZT0+SQRT(T(I))*RT2DZ
      TOF=T(I)-TDSC(NDSCS)
      CALL MOMENT(VGRAV,ZREF,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)
C
C     COMPUTE R THE RADIUS OF THE CLOUD
C
      ROH2=R2DSC(1)
      SIGW=SQRT(SIGW2+ROH2/2.)
      SIGP=SQRT(SIGP2+ROH2/2.)
      FR(I)=1.5*SQRT(SIGW*SIGP)
      ACL=CWIND(X,Y,ZREF,T(I))*OWF(NWL,NSOIL)
C
C     COMPUTE Z2 APPROXIMATE HEIGHT OF THE CLOUD
C
      FZ2(I)=(2.*QDSC(1,1))/PI/FR(I)/ACL
      Z22=FZ2(I)
C
C     COMPUTE THE X POSITION OF THE CLOUD AT A HEIGHT OF Z2 AND A HEIGHT
C     OF 1 METER.
C
      CALL MOMENT(VGRAV,Z22,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)
      XB(2)=XBAR+XDSC(1)
      Z1=1.0
      CALL MOMENT(VGRAV,Z1,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)
      XB(1)=XBAR+XDSC(1)
      FXC1(I)=(XB(2)-XB(1))/(Z22-Z1)
      FXC0(I)=XB(1)-FXC1(I)
   20 CONTINUE
C
C     COMPUTE AND STORE THE QUADRATIC FITS
C
      TIMES(ICOUNT)=T(3)
C
C     FIT AND STORE THE CLOUD RADIUS
C
      CALL FIT(T,FR,A,B,C)
      RT(1,ICOUNT)=A
      RT(2,ICOUNT)=B
      RT(3,ICOUNT)=C
C
C     FIT AND STORE Z2, APPROXIMATE CLOUD HEIGHT
C
      CALL FIT(T,FZ2,A,B,C)
      Z2(1,ICOUNT)=A
      Z2(2,ICOUNT)=B
      Z2(3,ICOUNT)=C
C
C     FIT AND STORE XC1
C
      CALL FIT(T,FXC1,A,B,C)
      XC1(1,ICOUNT)=A
      XC1(2,ICOUNT)=B
      XC1(3,ICOUNT)=C
C
C     FIT AND STORE XC0
C
      CALL FIT(T,FXC0,A,B,C)
      XC0(1,ICOUNT)=A
      XC0(2,ICOUNT)=B
      XC0(3,ICOUNT)=C
```

```
      TINC=1.2*TINC
      IF(ICOUNT.LT.20.AND.T(3).LT.TMAX) GO TO 10
      RETURN
      END
```

```
      SUBROUTINE RISE(TPRES,TNEXT,DEL)
      REAL M,NDIF,KZ,KX
      LOGICAL SWITCH,CHANGE
      DIMENSION WK(12,6)
      COMMON/BUOYCL/ Y(8),SPHNS(3),RISTIM
      COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,M,NDIF,ZINV
      COMMON /CLOCK/ TIME,TWIND
      COMMON/STARS/USTAR,TSTAR,ZSTAR
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/SIG/SIG02,SIGC
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
     1QDSC(20,3)
      COMMON/BURST/ACCEL,TBURST
      DATA HMIN,ACCURC,WK,N,ND/.001,.001,72*0.,8,12/
C ***************************************************************************
C
C
C     PURPOSE
C
C     THIS ROUTINE CALLS A RUNGA KUTTA ROUTINE TO INTEGRATE IN TIME
C     THE EQUATIONS FOR THE RISE OF A BUOYANT CLOUD BEGINNING AT TPRES
C     ENDING AT TNEXT UNLESS THE CONDITION FOR SWITCHING TO THE WIND
C     DISPERSION MODEL IS ENCOUNTERED IN WHICH CONVRT IS CALLED.
C     SEE SUBROUTINE DIFEQ FOR THE DEFINITIONS OF Y(I).
C
C
C      ARGUMENTS
C
C      TPRES      AS INPUT TPRES IS THE INITIAL TIME OF THIS SEGMENT OF
C                 INTEGRATION AND IS RETURNED WITH THE VALUE OF THE LAST
C                 SUCCESSFUL INTEGRATION STEP.
C      TNEXT      THE ENDTPOINT OF THE TIME INTERVAL WHICH IS INPUT.
C
C     REQUIRED SUBROUTINES
C
C          RKM      A RUNGA-KUTTA-MERSON INTEGRATION ROUTINE
C
C          CONVRT   A SUBROUTINE WHICH CONVERTS THE CURRENT BUOYANT
C                   DUST CLOUD TO A NUMBER OF DISC SOURCES FOR THE
C                   WIND DISPERSION MODEL.  A GAP TIME DURING WHICH THE
C                   BUOYANT MODEL IS CONTINUED IS COMPUTED.
C
C          WNDCAL   COMPUTES SCALED WIND SPEED AT A SPECIFIED HEIGHT
C
C          DIFFUS   COMPUTES DIFFUSIVITY AT A SPECIFIED HEIGHT
C
C     CALLED BY DUSTCL
C
C ***************************************************************************
      IF(TNEXT.GT.TWIND)GO TO 999
      SWITCH=.FALSE.
      CHANGE=.FALSE.
      T2=TPRES
C
C     PERFORM INTEGRATION IN SEGMENTS OF TIME
C
   10 DO 20 NT=1,300
      T1=T2
      T2=1.2*T1
      IF(T2.LE.0.)T2=.5
      IF(T2.GT.TNEXT)T2=TNEXT
      IF (DEL.LT.HMIN)DEL=HMIN
      CALL RKM(N,T1,T2,Y,HMIN,DEL,ACCURC,WK,ND)
C
C     CHECK TO SEE IF CLOUD GROWTH IS DOMINATED BY WIND DIFFUSION
C     OVER BUOYANT RISE BY COMPARING WIND DIFFUSIVITY, DIFW, TO
C     THE EFFECTIVE BUOYANT DIFFUSIVITY, DIFB AND IF THE HEIGHT
C     OF THE CENTER OF MASS IS LESS THAN ZSTAR SWITCH TO THE
```

```
C       WIND MODEL.
C
        IF(T2.LT.TBURST)GO TO 15
   5    DIFB=ABS(.1111*Y(1)*Y(3))
        TOP=Y(6)+Y(1)
        DIFW=DIFFUS(Z0,ZL,TOP)
        IF(DIFB.GT.DIFW)GO TO 15
        IF(TOP.GT.ZSTAR)SWITCH=.TRUE.
        IF(TOP.GT.ZSTAR)GO TO 15
        VTR=USTAR*WNDCAL(Z0,ZL,Y(6))
        KZ=DIFFUS(Z0,ZL,Y(6))
        KX=DXZ0*KZ
        TTR=T2
        XTR=Y(4)
        ZTR=Y(6)
        CHANGE=.TRUE.
        CALL CONVRT(T2)
        GO TO 200
  15    CONTINUE
        IF(T2.GE.TNEXT)GO TO 200
        IF(T2.GT.300.)GO TO 99
  20    CONTINUE
  99    WRITE(IOOUT,98)
  98    FORMAT(54H *** DIRTRAN ERROR - 5 MINUTE CUT-OFF ON BUOYANT RISE )
        STOP
 200    TPRES=T2
        RISTIM=TPRES
 999    RETURN
        END
```

```
      SUBROUTINE RKM(N,XL,XU,Y,HMIN,DEL,ACCURC,WK,ND)
C
C
C     NUMERICAL INTEGRATION ROUTINE FOR SYSTEMS OF ODE'S
C           USING THE RUNGE-KUTTA-MERSON TECHNIQUE
C
C
C     INPUT PARAMETERS
C
C         N - NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS
C        XL - INITIAL ABCISSA OF THE INTERVAL
C        XU - THE FINAL ABCISSA OF THE INTEGRATION INTERVAL
C         Y - A SINGLY DIMENSIONED ARRAY OF LENGTH N.  WHEN
C             RKM IS CALLED IT MUST CONTAIN THE VALUES OF
C             THE DEPENDENT VARIABLES AT XL.  UPON RETURN
C             TO THE CALLING PROGRAM Y CONTAINS THE VALUES
C             OF THE DEPENDENT VARIABLES AT XU.
C      HMIN - THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE
C             INTEGRATION
C       DEL - THE INITIAL ESTIMATE OF THE STEP SIZE AND UPON
C             RETURN TO THE CALLING PROGRAM DEL CONTAINS THE
C             FINAL STEP SIZE USED.  THIS VALUE SHOULD BE USED
C             IN THE NEXT CALL TO PRODUCE AN EFFICIENT INTEGRATION.
C             DEL IS RETURNED WITH THE VALUE ZERO IF IT HAS
C             BEEN HALVED BELOW HMIN.
C    ACCURC - PREASSIGNED ACCURACY WHICH IS ALSO USED IN ADJUSTING
C             THE STEP SIZE.
C        WK - AT LEAST A BLOCK OF N BY 6 FLOATING POINT LOCATIONS
C             USED FOR A WORK ARRAY.
C        ND - THE DIMENSION OF ARRAYS Y AND WK.
C
C
C        IT IS REQUIRED THAT THE USER OF RKM WRITE A SUBROUTINE
C     DEFINING THE DIFFERENTIAL EQUATIONS.  THE SUBROUTINE
C     STATEMENT SHOULD LOOK LIKE - SUBROUTINE DIFEQ(N,X,Y,YP) .
C
C     WHERE
C           N - THE NUMBER OF EQUATIONS
C           X - THE INDEPENDENT VARIABLE
C           Y - SINGLY DIMENSIONED ARRAY OF DEPENDENT VARIABLES
C          YP - SINGLY DIMENSIONED ARRAY OF THE RATES OF Y AT X
C                      YP(I) = D Y(I)/DX
C
C
      DIMENSION Y(ND),WK(ND,6)
      LOGICAL FIRST,QUIT
C
C     SET UP NEEDED VARIABLES UPON ENTRY
C
      XN=XL
      H=DEL
      FIRST=.TRUE.
      QUIT=.FALSE.
C
C     CHECK IF XN IS CLOSE TO XU
C
   20 IF(XN+H .LT. XU) GO TO 30
      DEL=H
      H=XU-XN
      QUIT=.TRUE.
      IF(FIRST) DEL=H
C
C     MAKE FIRST CALL TO DIFEQ AT THE BEGINNING OF INTERVAL
C
   30 CALL DIFEQ(N,XN,Y,WK(1,1))
C
C     PERFORM THE RUNGE-KUTTA-MERSON ALGORITHM
C
   40 H3=H/3.
      DO 50 I=1,N
```

```
      WK(I,3)=H3*WK(I,1)
   50 WK(I,6)=Y(I)+WK(I,3)
      CALL DIFEQ(N,XN+H3,WK(1,6),WK(1,2))
      DO 60 I=1,N
   60 WK(I,6)=Y(I)+(WK(I,3)+H3*WK(I,2))/2.
      CALL DIFEQ(N,XN+H3,WK(1,6),WK(1,2))
      DO 70 I=1,N
      WK(I,4)=H3*WK(I,2)
   70 WK(I,6)=Y(I)+(3.*WK(I,3)+9.*WK(I,4))/8.
      CALL DIFEQ(N,XN+H/2.,WK(1,6),WK(1,2))
      DO 80 I=1,N
      WK(I,5)=H3*WK(I,2)
   80 WK(I,6)=Y(I)+(3.*WK(I,3)-9.*WK(I,4)+12.*WK(I,5))/2.
      CALL DIFEQ(N,XN+H,WK(1,6),WK(1,2))
C
C  FIND THE LARGEST RELATIVE ERROR
C
      TEST=0.
      DO 90 I=1,N
      YX=Y(I)
      IF(YX .EQ. 0.) YX=ACCURC
      E=((WK(I,3)-9.*WK(I,4)/2.+4.*WK(I,5)-H3*WK(I,2)/2.)/5.)/YX
   90 TEST=AMAX1(TEST,ABS(E))
      FIRST=.FALSE.
      IF(TEST .LT. ACCURC) GO TO 100
C
C  IF THE LARGEST ERROR IS GREATER THAN ACCURC HALF THE STEP
C    SIZE AND TRY AGAIN.
C
      H=H/2.
      IF(H .LT. HMIN) GO TO 10
      QUIT=.FALSE.
      GO TO 40
C
C  TRUNCATION ERROR LESS THAN ACCURC, RESET THE Y ARRAY TO
C    SET UP FOR THE NEXT INTERVAL
C
  100 XN=XN+H
      DO 110 I=1,N
  110 Y(I)=Y(I)+(WK(I,3)+4.*WK(I,5)+H3*WK(I,2))/2.
C
C  CHECK FOR STEP SIZE DOUBLING.  DOUBLE IF LARGEST RELATIVE
C      ERROR IS 32 TIMES LESS THAN ACCURC.
C
      IF(.NOT.(TEST .GE. ACCURC/32. .OR. QUIT)) H=H+H
      IF(.NOT. QUIT) GO TO 20
      RETURN
C
C  THE VALUE OF H (DEL) IS LESS THAN THE SPECIFIED MINIMUM.
C    REPORT THIS AND ERROR OUT.
C
   10 CONTINUE
 1000 FORMAT('1   H BELOW HMIN'/'0     INTEGRATION ABORTED')
      DEL=0.
      RETURN
      END
```

```
      SUBROUTINE SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)
      DIMENSION NCHS(2),SRCBAS(2),SIDE1(2),SIDE2(2),TRNFRM(2,2),REF(2)
      COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +            COOR(2,200),TSTAG(200),DMMY(401)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C*********************************************************************
C
C     PURPOSE
C
C       TO CONVERT THE USER DEFINED COORDINATES OF THE CHARGES TO THE
C       LOCAL COORDINATE SYSTEM.
C
C      INPUTS
C
C         NCHS - SINGLY DIMENSIONED ARRAY CONTAINING THE NUMBER OF
C                CHARGES.
C
C         SRCBAS - A REFERENCE CHARGE IN THE USER DEFINED COORDINATES.
C
C         SIDE1,SIDE2 - VECTORS DESCRIBING THE BOUNDING PARRALLELOGRAM.
C
C         TRNFRM - COORDINATE SYSTEM TRANSFORMATION MATRIX.
C
C    OUTPUTS  RETURNED IN COMMON /ARRAY/ AND /SEPRTN/
C
C        DIFF- DOUBLY DIMENSIONED ARRAY CONTAINING THE CHARGE COORDINATES
C              IN THE LOCAL COORDINATE SYSTEM.
C
C      ITOT- TOTAL NUMBER OF CHARGES.
C
C      NCHTOT- WHEN NARY=1 OR 2 THE TOTAL NUMBER OF CHARGES. WHEN
C              WHEN NARY=3 IS SET =1.
C
C      NTOT- WHEN NARY=1 OR 2 IS SET =1 AND WHEN NARY=3 IS THE TOTAL
C            NUMBER OF CHARGES.
C
C
C    SUBROUTINES AND FUNCTIONS
C
C      UNIT   COMPUTES THE UNIT VECTOR OF A GIVEN VECTOR
C*********************************************************************
      IF(NARY.NE.1)GO TO 4
      NCHTOT=NCHS(1)*NCHS(2)
      NTOT=1
      ITOT=NCHTOT
      GO TO 6
    4 IF(NARY.NE.2)GO TO 5
      NCHTOT=NCHS(1)
      NTOT=1
      ITOT=NCHTOT
      GO TO 6
    5 IF(NARY.NE.3)GO TO 998
      NCHTOT=1
      NTOT=NCHS(1)
      ITOT=NTOT
    6 CONTINUE
C
C     DETERMINE THE COORDINATE OF THE REFERENCE CHARGE IN THE INTERNAL
C     COORDINATE SYSTEM
C
      DO 20 I=1,2
      REF(I)=0.0
      DO 10 J=1,2
      REF(I)=REF(I)+TRNFRM(I,J)*SRCBAS(J)
   10 CONTINUE
   20 CONTINUE
      DO 40 I=1,2
      DO 30 J=1,ITOT
```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

3 - 6

```
      DIFF(I,J)=0.0
   30 CONTINUE
   40 CONTINUE
      IF(NARY.GT.1)GO TO 90
      NM=0
      NC1=NCHS(1)
      NC2=NCHS(2)
C
C  COMPUTE LOCATIONS OF CHARGES FOR INTERNAL COORDINATE SYSTEM FOR
C  UNIFORMLY DISTRIBUTED CHARGES.
C
      DO 80 M=1,NC2
      DO 70 N=1,NC1
      NM=NM+1
      DO 60 I=1,2
      DO 50 J=1,2
      DIFF(I,NM)=DIFF(I,NM)+FLOAT(N-1)*TRNFRM(I,J)*SIDE1(J)
     1 +FLOAT(M-1)*TRNFRM(I,J)*SIDE2(J)
   50 CONTINUE
   60 CONTINUE
   70 CONTINUE
   80 CONTINUE
      GO TO 999
   90 CONTINUE
C
C  TRANSFORM CHARGE LOCATIONS TO LOCAL COORDINATE SYSTEM FOR RANDOM
C  CHARGES.
C
      NC1=NCHS(1)
      DO 120 M=1,NC1
      DO 110 I=1,2
      DO 100 J=1,2
      DIFF(I,M)=DIFF(I,M)+TRNFRM(I,J)*COOR(J,M)
  100 CONTINUE
      DIFF(I,M)=DIFF(I,M)-REF(I)
  110 CONTINUE
  120 CONTINUE
      GO TO 999
  998 WRITE(IOOUT,778)
  778 FORMAT(5X,23H *** NARY OUT OF RANGE   )
  999 RETURN
      END
```

```
      SUBROUTINE SOURCE(W,NCHRG,DD,NSOIL,DSOD)
C     ****************************************************
C
C
C     PURPOSE
C
C     TO CALCULATE EXPLOSIVE DUST SOURCE TERM FOR THE
C               DIRTRAN CODE
C
C     INPUT
C
C       W        THE WEIGHT OF THE CHARGE IN KG-TNT
C       DD       DETONATION DEPTH IN METERS
C       NSOIL    INTEGER SOIL INDEX
C       DSOD     DEPTH OF SOD IN METERS
C
C     OUTPUT  (RETURNED IN COMMON /PRTINF/ , /BUOYCL/ AND /CARB/
C
C       R0       INITIAL CLOUD RADIUS IN METERS
C       VGRAV    SINGLY DIMENSIONED ARRAY CONTAINING OPTICALLY WEIGHTED
C                AVERAGE SETTLING VELOCITIES FOR EACH SIZE RANGE IN
C                THE PARTICLE DISTRIBUTION  (METERS/SEC)
C       NPRTS    THE NUMBER OF SIZE RANGES IN THE PARTITIONING OF THE
C                PARTICLE SIZE SPECTRUM
C       RSPH     THE INITIAL RADIUS OF THE CLOUD IN METERS
C       DELT     THE INITIAL DIFFERENCE IN TEMPERATURE BETWEEN THE CLOUD
C                AND SURROUNDINGS  (DEGREES KELVIN)
C       VZSPH    THE INITIAL VERTICAL VELOCITY OF THE CLOUD (M/S)
C       XCMSPH   INITIAL HORIZONTAL POSITION OF THE CLOUD (METERS)
C       YCMSPH   INITIAL Y POSITION OF THE CLOUD (METERS)
C       ZCMSPH   INITIAL HEIGHT OF THE CLOUD (METERS)
C       XTOP     INITIAL X POSITION OF THE TOP OF THE CLOUD (METERS)
C       YTOP     INITIAL Y POSITION OF THE TOP OF THE CLOUD (METERS)
C       RISTIM   TIME LAPSED SINCE DETONATION IN SECONDS
C
C       RCARB1   PORTION OF BUOYANT CLOUD WHICH IS DIRT PARTICLES
C
C       RCARB2   PORTION OF BUOYANT CLOUD WHICH IS CARBON PARTICLES
C
C
C     CALLED BY DUSTCL
C
C     SUBROUTINES AND FUNCTIONS
C
C     NONE
C
C     ***********************************************************
      LOGICAL HORIZ,ONCE
      DIMENSION CR(5,7),CD(5,7),OWML(3,4),OWSV(3,4),PRTTN(4)
      DIMENSION S(3),BURHTR(5),WTRAT(5)
      COMMON/PRTINF/ R0,VGRAV(3),NPRTS
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/BUOYCL/ RSPH,DELT,VZSPH,XCMSPH,YCMSPH,ZCMSPH,XTOP,YTOP,
     * SPHNS(3),RISTIM
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
      COMMON/STARS/USTAR,TSTAR,ZSTAR
      COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
      COMMON /BURST/ ACCEL,TBURST
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON /MODE/ HORIZ
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
     1 QDSC(20,3)
      COMMON/CARB/RCARB1,RCARB2
      COMMON/NTAL/TNOT,VOLSPH,TNO,CBLEED
C
C  CR IS THE CRATER RADIUS INDEXED BY COEFFICIENT AND SOIL TYPE
C
      DATA CR/.271,-.684,.39,.886,0.,.271,-.684,.39,.886,0.,
     1 .386,-.849,.367,.993,0.,.503,-.954,.45,1.19,0.,
     2 .629,-1.08,.264,1.12,0.,.629,-1.08,.264,1.12,0.,
```

```
     3 .806,-1.28,-.178,.852,0./
C
C  CD IS THE CRATER DEPTH INDEXED BY COEFFICIENT AND SOIL TYPE
C
      DATA CD/.113,-.477,.27,1.84,1.05,.134,-.571,.343,2.24,1.31,
     1 .189,-.84,.447,3.3,2.1,.251,-1.17,.494,4.72,3.34,
     2 .189,-.84,.447,3.3,2.1,.331,-1.49,.579,4.92,3.13,
     3 .449,-1.82,.322,4.11,2.02/
C
C  OWML IS THE OPTICALLY WEIGHTED MASS LOADING COEFFICIENT INDEXED BY
C       BIN SIZE AND SOIL TYPE
C
      DATA OWML/2.88E3,2*0.,3.08E3,8*0./
C
C  OWSV IS THE OPTICALLY WEIGHTED PARTICLE SETTLING VELOCITY (CM/SEC)
C       INDEXED BY BIN SIZE AND SOIL TYPE
C
      DATA OWSV/12*0./
C
C  PRTTN IS THE PARTITIONING RATIO INDEXED ON SOIL TYPE
C
      DATA PRTTN/4*.8/
C
C   BURHTR IS THE RATIO OF BURST HEIGHT TO INITIAL RADIUS AND WTRAT
C   IS THE FRACTION OF THE TOTAL WEIGHT WHICH IS EFFECTIVE IN THE CLOUD
C
      DATA BURHTR/0.,4.,2.,4.,3./,WTRAT/.6,1.,.8,1.,.7/
C
C
      RISTM=0.
      XCMSPH=0.
      YCMSPH=0.
      XTOP=0.
      YTOP=0.
      TNO=T0
      NPRTS=1
C   SCARB IS THE OPTICALLY WEIGHTED CARBON PARTICLE LOADING COEFFICIENT
      SCARB=270.*W
      W3=(W*WTRAT(NCHRG))**.3333333
      R0=2.0*W3
      TAMB=T0+TMPCAL(Z0,ZL,R0)*TSTAR
      DELT=.57*TAMB
      RSPH=R0
      ZCMSPH=R0
      BURHT=BURHTR(NCHRG)*R0
      BURVZ=1.3*SQRT(R0)
      TBURST=.15*R0
      VZSPH= 2.*BURHT/TBURST-BURVZ
      ACCEL=(BURVZ-VZSPH)/TBURST
      VOLSPH=(4./3.)*3.141593*R0**3
      TNOT=T0+DELT
      CLAM=DD/W3
C
C  CALCULATE CRATER RADIUS AND DEPTH
C
      ONCE=.FALSE.
      IF(NSOIL.EQ.1)IDX=4
      IF(NSOIL.EQ.2.)IDX=3
      GO TO 70
   60 IF(NSOIL.EQ.1)IDX=6
      IF(NSOIL.EQ.2)IDX=4
   70 CONTINUE
      RC=CR(1,IDX)
      DC=CD(1,IDX)
      IF (CLAM.LT.1.E-30) GO TO 98
      TERM=1.
      DO 100 I=2,5
      TERM=TERM*CLAM
      RC=RC + CR(I,IDX)*TERM
      DC=DC + CD(I,IDX)*TERM
```

```
100   CONTINUE
   98 CONTINUE
      RC=RC*W3
      DC=DC*(W*WTRAT(NCHRG))**.3
C
C  GET CRATER VOLUME
C
      DSDC=DSOD/DC
      VC=(2.*3.141592/3.)*RC*RC*DC*(1.-1.5*DSDC*(1.-DSDC*DSDC/3.))
      IF(DSOD.GE.DC)VC=0.0
      IF(ONCE)GO TO 110
      ONCE=.TRUE.
      IF(NSOIL.EQ.1)VC1=.5*VC
      IF(NSOIL.EQ.2)VC1=.25*VC
      GO TO 60
  110 IF(NSOIL.EQ.1)VC=VC1+.5*VC
      IF(NSOIL.EQ.2)VC=VC1+.75*VC
C
C  CALCULATE OPTICALLY WEIGHTED PARAMETERS
C
      NDSCS=MIN0(10,IFIX(5.*W3/1.8))
      CBLEED=0.
      DO 101 L=1,NPRTS
      S(L)=QWML(L,NSOIL) * VC
      VGRAV(L)=QWSV(L,NSOIL)
      SPHNS(L)=PRTTN(NSOIL) * S(L)
      QDSC(1,L) =(1.-PRTTN(NSOIL)) * S(L)/FLOAT(NDSCS)
      CBLEED=CBLEED+S(L)
101   CONTINUE
      CBLEED=CBLEED*.03/W3**3
      RCARB=SCARB/SPHNS(1)
      RCARB1=1./(1.+RCARB)
      RCARB2=RCARB/(1.+RCARB)
      SPHNS(1)=SPHNS(1)+SCARB
      DELH=2.*R0/FLOAT(NDSCS)
      Z=-DELH/2.
      DO 200 I=1,NDSCS
      Z=Z+DELH
      ZDSC(I)=Z
      DO 201 J=1,NPRTS
      QDSC(I,J)=QDSC(1,J)
  201 CONTINUE
      CON=ALOG(QDSC(I,1)/VISEXT/DELH/(2.*R0)/3.14159)
      IF(CON.GT.1.)GO TO 210
      D=1.
      GO TO 230
  210 D=CON
      DO 220 IT=1,5
      D=(CON-1.+ALOG(D))*D/(D-1)
  220 CONTINUE
  230 R2DSC(I)=4.*R0*R0/D
      TDSC(I)=-DELH*DELH/D/(DZ0*Z**DN)/4.
      SIGZ=DELH*DELH/D
      XDSC(I)=U0*Z**UM * TDSC(I)
  200 CONTINUE
  999 RETURN
      END
```

```
      SUBROUTINE TEMP(Z,TA,DTADZ)
C
C    PURPOSE
C        TO COMPUTE THE AMBIENT ATMOSPHERIC POTENTIAL TEMPERATURE AND
C        GRADIENT AT A GIVEN HEIGHT.
C
C    INPUTS
C
C        Z    HEIGHT AT WHICH AMBIENT TEMPERATURE AND TEMPERATURE
C             GRADIENT ARE DESIRED.
C
C   OUTPUTS
C
C     TA      AMBIENT POTENTIAL TEMPERATURE
C
C     DTADZ   TEMPERATURE GRADIENT
C
C    SUBROUTINES AND FUNCTIONS NEEDED
C
C        TMPCAL    COMPUTES SCALED TEMPERATURE AT A GIVEN HEIGHT
C
C    CALLED BY DIFEQ, ATMCAL
C
C*********************************************************************
      COMMON/STARS/USTAR,TSTAR,ZSTAR
      COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT
      COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      S=Z/ZL
      TA=TSTAR*TMPCAL(Z0,ZL,Z)+T0
      IF(ABS(ZL).LE.1.E03)GO TO 10
C
C  NEUTRAL CASE
C
      DTADZ=TSTAR/(Z0+Z)
      GO TO 999
   10 IF(ZL.GT.0.0)GO TO 15
C
C  UNSTABLE CASE
C
      DTADZ=(TSTAR/Z)*(1.-16.*(S))**(-1./2.)
      IF(S.LT.-2.0)DTADZ=(TSTAR/ZL)*(AT/3.*(-ZL/Z)**(4./3.))
      GO TO 999
C
C  STABLE CASE
C
   15 DTADZ=(TSTAR/ZL)*(ZL/(Z0+Z)+11.)
      IF(S.GT.1.5)DTADZ=BT*TSTAR/ZL
  999 RETURN
      END
```

```
      FUNCTION TMPCAL(Z0,ZL,Z)
C ***********************************************************************
C
C
C
C     PURPOSE
C
C     TO CALCULATE THE POTENTIAL TEMPERATURE SCALED BY THE SCALE
C     TEMPERATURE, T*, FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV
C     LENGTH AT A SPECIFIED HEIGHT.
C
C
C     INPUTS
C
C     Z0       THE FRICTION HEIGHT IN METERS.
C     ZL       THE MONIN-OBUKHOV LENGTH IN METERS.
C     Z        THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED
C              IN METERS
C
C
C     RETURNS SCALED TEMPERATURE
C
C
C     CALLED BY ATMCAL, SOURCE AND TEMP
C
C ***********************************************************************
      LOGICAL LOW
      COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      PHIM(Z)=(1.-16.*Z)**(-.25)
      PSIH(S,S0)=ALOG(((S**2.-1.)/(S**2.+1.))*((S0**2.+1.)/(S0**2.-1.)))
      PSIHS(Z)=-11.*Z
C     TRACE 999
C
C
C     PHIM      THE SHEAR OF MOMENTUM
C     PSIH      THE UNIVERSAL FUNCTION FOR DEVIATION FROM LOGARITHMIC
C               POTENTIAL TEMPERATURE PROFILE IN THE BOUNDARY LAYER
C               OF AN UNSTABLE ATMOSPHERE
C     PHIHS     THE SAME AS PHIH EXCEPT FOR STABLE ATMOSPHERE
C
      IF(ABS(ZL).LE.1.E3)GO TO 100
      TMPCAL=ALOG(1.+Z/Z0)
      GO TO 999
  100 CONTINUE
      P=SIGN(1.,ZL)
      LOW=.TRUE.
      S=Z/ZL
      IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10
      S=AMIN1(S,1.5)
      S=AMAX1(S,-2.)
      LOW=.FALSE.
   10 CONTINUE
      IF(P)120,130,130
  120 S=1./PHIM(S)
      S1=Z0/ZL
      S0=1./PHIM(S1)
      TMPCAL=PSIH(S,S0)
C
C  FIND CONSTANTS FOR MATCHING IN UNSTABLE CASE AT Z/ZL=-2.
C
      S2=-2.
      AT=-3.*((1.-16.*S2)**(-1./2.))*(-S2)**(1./3.)
      CT=-1.*AT*(-S2)**(-1./3.)
      GO TO 52
  130 CONTINUE
      PSI=PSIHS(S)
      TMPCAL=ALOG(1.+S*ZL/Z0)-PSI
C
C  FIND CONSTANTS FOR MATCHING IN STABLE CASE AT Z/ZL=1.5
```

```
C
      S2=1.5
      BT=1./(Z0/ZL+S2)+11.
      DT=-1.*BT*S2
   52 CONTINUE
      IF(LOW)GO TO 999
      IF(P)53,53,54
   53 TMPCAL=TMPCAL+CT+AT*(-ZL/Z)**(1./3.)
      GO TO 999
   54 TMPCAL=TMPCAL+DT+BT*Z/ZL
  999 RETURN
      END
```

```
      SUBROUTINE TRAP(TRSK,TRWK,TRSP,H,SIGW,SIG0,TIME,SKT,WAK,SPH)
C
C  THIS SUBROUTINE PERFORMS A TRAPEZOID INTEGRATION
C
C  INPUTS
C
C     TRSK  - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT
C               TO THE CENTER OF THE SKIRT
C
C     TRWK  - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT
C               TO THE CENTER OF THE WAKE
C
C     TRSP  - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT
C               TO THE CENTER OF THE SPHERE
C
C     H     - INTEGRATION STEP SIZE THROUGH THE SKIRT
C
C     SIGW  - INTEGRATION STEP SIZE THROUGH THE WAKE
C
C     SIG0  - INTEGRATION STEP SIZE THROUGH THE SPHERE
C
C     TIME  - TIME TRANSMITTANCE IS DESIRED
C
C  ALL OTHER NEEDE INFORMATION IS PASSED VIA COMMON BLOCKS
C
C
C  OUTPUTS
C
C     SKT  - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF
C              SIGHT FROM THE SKIRT
C
C     WAK  - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF
C              SIGHT FROM THE WAKE ONCE THE BUOYANT SPHERE HAS
C              CONVERTED TO THE WIND MODEL
C
C     SPH  - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF
C              SIGHT FROM THE SPHERE ONCE IT HAS CONVERTED TO THE
C              WIND MODEL
C
C  FUNCTIONS NEEDED
C
C     CWIND  - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT
C                (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SKIRT
C
C     CWAKE  - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT
C                (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE WAKE
C
C     CSPHER - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT
C                (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SPHERE
C
C*********************************************************************
      REAL KX,KZ
      LOGICAL SWITCH,CHANGE,ONCE
      DIMENSION OWF(5,2),OWFC(5),TRSK(3),TRWK(3),TRSP(3)
      COMMON/LOS/TR(3),RE(3),U(3)
      COMMON/ACL/CWINDS,CWINDC,CWINDW
      COMMON/CARB/RCARB1,RCARB2
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/SIG/SIG02,SIGC
      COMMON/POINTS/XNORM,DOT1,DOT2,DOT3
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
      DATA OWFC/1.,.95,.5,.2,1.E-03/
      TI=TIME
      SKT=0.0
      WAK=0.0
      SPH=0.0
      SUM=0.0
      SUM1=0.0
      SUM2=0.0
      IND=20
```

```
      ONCE=.FALSE.
      FP=0.0
      F=0.0
      J=0
      DO 10 I=1,IND
      XI=H*FLOAT(I-1)
      DIST=DOT1+XI
      IF(DIST.GT.XNORM)GO TO 10
      IF(DIST.LT.0.0)GO TO 10
      X=TRSK(1)+XI*U(1)
      Y=TRSK(2)+XI*U(2)
      Z=TRSK(3)+XI*U(3)
      FP=F
      ACL=CWIND(X,Y,Z,TI)
      F=ACL
C
C  CHECK TO SEE IF THE CONTRIBUTION IS NEGLIGABLE
C
      IF(.NOT.ONCE)ACL1=ACL
      IF(ACL1.LT.1.E-10)GO TO 11
      PER=.01*ACL1
      ONCE=.TRUE.
      J=J+1
      IF(ACL.LT.PER)GO TO 11
      IF(J.LE.1)GO TO 10
      SUM=SUM+FP+F
   10 CONTINUE
   11 CONTINUE
      IF((TIME-TTR).LT.1.E-20)GO TO 31
C
C  COMPUTE THE CONTRIBUTION TO CL FROM THE SPHERE USING A TRAPEZOID
C  INTEGRATION ONLY AFTER THE BUOYANT FIREBALL HAS CONVERTED TO THE WIND
C  MODEL
C
      ONCE=.FALSE.
      FP=0.0
      F=0.0
      J=0
      DO 20 I=1,IND
      XI=SIG0*FLOAT(I-1)
      DIST=DOT3+XI
      IF(DIST.GT.XNORM)GO TO 20
      IF(DIST.LT.0.0)GO TO 20
      X=TRSP(1)+XI*U(1)
      Y=TRSP(2)+XI*U(2)
      Z=TRSP(3)+XI*U(3)
      FP=F
      ACL=CSPHER(X,Y,Z,TI)
      F=ACL
      IF(.NOT.ONCE)ACL1=ACL
      IF(ACL1.LT.1.E-05)GO TO 21
      PER=.01*ACL1
      ONCE=.TRUE.
      J=J+1
      IF(ACL.LT.PER)GO TO 21
      IF(J.LE.1)GO TO 20
      SUM1=SUM1+FP+F
   20 CONTINUE
   21 CONTINUE
C
C  COMPUTE CONTRIBUTION TO CL FROM THE WAKE AFTER THE BUOYANT FIREBALL
C  HAS CONVERTED TO THE WIND MODEL USING TRAPEZOID INTEGRATION WITH STEP
C  SIZE SIGW.
C
      ONCE=.FALSE.
      FP=0.0
      F=0.0
      J=0
      DO 30 I=1,IND
      XI=SIGW*FLOAT(I-1)
```

```
      DIST=DOT2+XI                                                      TRP01410
      IF(DIST.GT.XNORM)GO TO 30                                         TRP01420
      IF(DIST.LT.0.0)GO TO 30                                           TRP01430
      X=TRWK(1)+XI*U(1)                                                 TRP01440
      Y=TRWK(2)+XI*U(2)                                                 TRP01450
      Z=TRWK(3)+XI*U(3)                                                 TRP01460
      FP=F                                                              TRP01470
      ACL=CWAKE(X,Y,Z,TI)                                               TRP01480
      F=ACL                                                             TRP01490
      IF(.NOT.ONCE)ACL1=ACL                                             TRP01500
      IF(ACL1.LT.1.E-05)GO TO 31                                        TRP01510
      PER=.01*ACL1                                                      TRP01520
      ONCE=.TRUE.                                                       TRP01530
      J=J+1                                                             TRP01540
      IF(ACL.LT.PER)GO TO 31                                            TRP01550
      IF(J.LE.1)GO TO 30                                                TRP01560
      SUM2=SUM2+FP+F                                                    TRP01570
   30 CONTINUE                                                          TRP01580
   31 CONTINUE                                                          TRP01590
      SKT=(ABS(H)/2.)*SUM                                               TRP01600
      WAK=(ABS(SIGW)/2.)*SUM2                                           TRP01610
      SPH=(ABS(SIGO)/2.)*SUM1                                           TRP01620
  999 RETURN                                                            TRP01630
      END                                                               TRP01640
```

```
      SUBROUTINE TRNCAL(TRN,REC,TIME,TRNLOS)
C
C   CONTROLING ROUTINE FOR CALCULATING TRANSMITTANCES FOR CHARGE
C   DISTRIBUTION TYPES 1 AND 2
C
C   INPUTS
C
C   TRN    - TRANSMITTER COORDINATES IN THE LOCAL COORDINATE SYSTEM
C
C   REC    - RECEIVER COORDINATES IN THE LOCAL COORDINATE SYSTEM
C
C   TIME   - TIME AFTER THE DETOATION AT WHICH A TRANSMITTANCE IS DESIRED
C
C    ALL OTHER NECESSARY INPUTS ARE PASSED IN COMMON BLOCKS
C
C
C   OUTPUT
C
C     TRNLOS   - TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT
C
C   SUBROUTINES NEEDED
C
C      AVRG   - FINDS THE AVERAGE OF THE MOMENTS FOR THE DISCS
C
C      VSUM   - ADDS TWO VECTORS
C
C      UNIT   - DETERMINE A UNIT VECTOR
C
C      TRNCLD- DETERMINE THE LENGTH OF THE INTERSECTION OF THE LINE OF
C              SIGHT WITH THE WAKE AND SPHERE
C
C      TRAP   - DOES A TRAPAZOIDAL INTEGRATION THROUGH SKIRT WAKE AND S
C              SPHERE FOR NON-HORIZONTAL LINES OF SIGHT
C
C      TRNCHK- CHECKS TO SEE IF THE OBSCURATION IS SUCH THAT THE
C              TRANSMITTANCE IS LESS THAN A SPECIFIED VALUE
C
C
C   FUNCTIONS NEEDED
C
C      DOTPRD - FINDS THE DOTPRODUCT OF TWO VECTORS
C
C      CWIND  - FINDS THE CONCENTRATION ALONG A SPECIFIED HORIZONTAL
C               LINE OF SIGHT OR DETERMINES THE CONCENTRATION AT SOME
C               POINT ALONG THE LINE OF SIGHT FROM THE SKIRT
C
C      CWAKE  - SAME AS CWIND EXCEPT FOR WAKE
C
C      CSPHER - SAME AS CWIND EXCEPT FOR BUOYANT SPHERE
C
C***********************************************************************
C
      REAL KZ,KX
      DIMENSION TRN(3),REC(3),OWF(5,2),OWFC(5),TEMP(2)
      DIMENSION DIR(2),                   XW(3),XS(3)
      DIMENSION TRSK(3),TRWK(3),TRSP(3)
      LOGICAL HORIZ,SWITCH,CHANGE,TEST,SKIP
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/CARB/RCARB1,RCARB2
      COMMON/BUOYCL/RSPH,DELT,YZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +            DMM(600),DMMY(401)
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON/MODE/HORIZ
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
      COMMON/ACL/CWINDS,CWINDC,CWINDW
      COMMON/LOS/T(3),R(3),U(3)
      COMMON/SIG/SIG02,SIGC
      COMMON/CHARGE/NCHG
      COMMON/POINTS/XNORM,DOT1,DOT2,DOT3
```

```
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
      COMMON/SKIPIT/SKIP
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
      DATA ONEM/-1.0/
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
      DATA OWFC/1.,.95,.5,.2,1.E-03/
C
C     PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER
C
      NCHG=1
      SKIP=.FALSE.
      TEST=.FALSE.
      HSPH=0.0
      HWAK=0.0
      ACLSKT=0.0
      ACLWAK=0.0
      ACLSPH=0.0
      XNORM=0.0
      DO 10 I=1,3
      R(I)=REC(I)
      T(I)=TRN(I)
      U(I)=R(I)-T(I)
      XNORM=XNORM+U(I)**2
   10 CONTINUE
      XNORM=SQRT(XNORM)
      U(1)=U(1)/XNORM
      U(2)=U(2)/XNORM
      U(3)=U(3)/XNORM
      IF((TIME-TTR).LT.1.E-20)GO TO 14
C
C     IF THE BUOYANT SPHERE HAS BECOME WIND BLOWN DETERMINE THE CENTER OF
C     MASS OF THE REFERENCE CHARGE.
C
      XCM=XTR+VTR*(TIME-TTR)
      ZCM=ZTR
C
C     DETERMINE CENTER OF WAKE FOR REFERENCE CHARGE
C
   14 ZX=5.0
      CALL AVRG(ZX,TIME,QTOT,XBAVRG,SIG2X,SIG2Y)
      XW(1)=(XBAVRG+XCM)/2.
      XW(2)=YCM/2.
      XW(3)=(5.0+ZCM)/2.
C
C     IF THE DIFFERENCE BETWEEN THE TRANSMITTER AND RECEIVER IS GREATER
C     THAN 1 PERCENT OF THE DISTANCE BETWEEN THEM THEN THE LOS IS
C     CONSIDERED A SLANT PATH.
C
   13 IF(ABS(U(3)).GT..01)GO TO 20
      IF(ABS(TRN(3)-REC(3)).LT.1.E-06)GO TO 9
C
C     COMPUTE CLOSEST POINT ALONG THE LOS TO OUR ESTIMATE OF THE CENTER OF
C     THE WAKE
C
      DOT=-(U(1)*(T(1)-XW(1))+U(2)*(T(2)-XW(2))+U(3)*(T(3)-XW(3)))
      T(3)=T(3)+U(3)*DOT
      R(3)=T(3)
C
C     COMPUTE CONTRIBUTIONS FOR A HORIZONTAL PATH
C
    9 HORIZ=.TRUE.
      CALL VSUM(REC,TRN,ONEM,DIR)
      CALL UNIT(DIR,DIR,RANGE)
      COSTH=DIR(1)
      SINTH=DIR(2)
      SINTH2=SINTH*SINTH
      COSTH2=COSTH**2
      SCRN(1)=SINTH
      SCRN(2)=-COSTH
      DO 12 J=1,ITOT
```

```
      DO 11 I=1,2
      TEMP(I)=DIFF(I,J)
   11 CONTINUE
      PRSEP(J)=DOTPRD(TEMP,SCRN)
   12 CONTINUE
      X=DOTPRD(SCRN,TRN)
C
C   COMPUTE THE CONTRIBUTION FROM THE SKIRT AT A HEIGHT OF T(3) WHERE T(3
C   IS THE HEIGHT OF THE TRANSMITTER IF THE DIFFERENCE BETWEEN THE
C   TRANSMITTER AND RECEIVER HEIGHTS IS SMALL AND IS THE Z COMPONENT OF
C   THE POINT ON THE LINE CONNECTING THE TRNSMITTER AND RECEIVER ,WHICH
C   IS CLOSEST TO OUR ESTIMATE OF THE CENTER OF THE WAKE OTHERWISE.
C
      ACLSKT=CWIND(X,Y,T(3),TIME)
      IF(TEST)GO TO 998
C
C   TEST IS A LOGICAL VARIABLE RETURNED IN COMMON/TRANNY/ FROM SUBROUTINE
C   TRNCHK WHICH IS CALLED BY CWIND, CWAKE, CSPHER, TRNCLD, AND TRAP
C   EACH TIME A CONTRIBUTION IS MADE TO THE OPTICALL WEIGHTED
C   CONCENTRATION ALONG THE OPTICAL PATH
C
C           TEST=.FALSE.  TRANSMITTANCE IS GREATER THAN TRNMIN
C                         (A TRANSMITTANCE THRESHOLD).
C               =.TRUE.   TRANSMITTANCE IS LESS THAN TRNMIN
C
      IF(TIME.GT.TTR)ACLWAK=CWAKE(X,Y,T(3),TIME)
      IF(TEST)GO TO 998
      IF(TIME.GT.TTR)ACLSPH=CSPHER(X,Y,T(3),TIME)
      IF(TEST)GO TO 998
      IF(TIME.GT.TTR)GO TO 50
      CWINDS=ACLSKT
      CWINDW=ACLWAK
      CWINDC=ACLSPH
      CALL TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)
      IF(TEST)GO TO 998
      GO TO 50
C
C   DO TRAPEZOIDAL INTEGRATION FOR SLANT PATH IN BOTH DIRECTIONS FROM
C   AN ESTIMATE OF THE LOCATION OF CENTER OF THE SKIRT USING A STEP SIZE
C   OF SIG,THE GEOMETRIC MEAN OF THE AVERAGE OF THE SPREADS OF THE DISCS
C   IN BOTH THE X AND Y DIRECTION.THEN IF THE  BUOYANT SPHERE HAS
C   CONVERTED TO THE WIND MODEL DO THE SAME FOR THE WAKE AND SPHERE
C   WITH THE APPROPRIATE STEP SIZE. (CHECK TO SEE IF SPHERE HAS
C   CONVERTED TO THE WIND MODEL IS DONE IN TRAP.)
C
   20 HORIZ=.FALSE.
      SIGX=SQRT(SIG2X)
      SIGY=SQRT(SIG2Y)
      SIG=SQRT(SIGX*SIGY)
      CALL WIN(2.0,UX,UY)
      XS(1)=TIME*UX
      XS(2)=TIME*UY
      XS(3)=2.0
C
C   FIND THE POINTS ON THE LINE CONNECTING THE TRANSMITTER AND
C   RECEIVER THAT ARE CLOSEST TO OUR ESTIMATE OF THE CENTER OF
C   THE SKIRT,WAKE, AND SPHERE.
C
      DO 48 J=1,ITOT
      XSK=XS(1)+DIFF(1,J)
      YSK=XS(2)+DIFF(2,J)
      ZSK=XS(3)
      DOT1=-(U(1)*(T(1)-XSK)+U(2)*(T(2)-YSK)+U(3)*(T(3)-ZSK))
      IF(DOT1.LT.0.0)DOT1=0.0
      IF(DOT1.GT.XNORM)DOT1=XNORM
      XWK=XW(1)+DIFF(1,J)
      YWK=XW(2)+DIFF(2,J)
      ZWK=XW(3)
      DOT2=-(U(1)*(T(1)-XWK)+U(2)*(T(2)-YWK)+U(3)*(T(3)-ZWK))
      IF(DOT2.LT.0.0)DOT2=0.0
```

```
      IF(DOT2.GT.XNORM)DOT2=XNORM
      XSP=XCM+DIFF(1,J)
      YSP=YCM+DIFF(2,J)
      ZSP=ZCM
      DOT3=-(U(1)*(T(1)-XSP)+U(2)*(T(2)-YSP)+U(3)*(T(3)-ZSP))
      IF(DOT3.LT.0.0)DOT3=0.0
      IF(DOT3.GT.XNORM)DOT3=XNORM
      NCHG=J
      DO 45 II=1,3
      TRSK(II)=T(II)+U(II)*DOT1
      TRWK(II)=T(II)+U(II)*DOT2
      TRSP(II)=T(II)+U(II)*DOT3
   45 CONTINUE
      DIFS=DIFFUS(Z0,ZL,2.0)
      SIGZ=SQRT(2.*DIFS*TIME)
      IF(SIGZ.LT.1.0)SIGZ=1.0
      H=U(1)*SIGX+U(2)*SIGY+U(3)*SIGZ
      IF(TIME.LT.TTR)GO TO 46
      HSPX=SQRT(SIG02+2.*KX*(TIME-TTR))
      HSPY=HSPX
      HSPZ=SQRT(SIG02+2.*KZ*(TIME-TTR))
      HSPH=(U(1)*HSPX+U(2)*HSPY+U(3)*HSPZ)/2.
      HWAK=(HSPH+H)/2.
   46 CONTINUE
      CALL TRAP(TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH)
      ACLSKT=ACLSKT+SKT
      ACLWAK=ACLWAK+WAK
      ACLSPH=ACLSPH+SPH
      CALL TRNCHK(ACLSKT,ACLWAK,ACLSPH)
      IF(TEST)GO TO 998
      H=-H
      IF(TIME.LT.TTR)GO TO 47
      HSPH=-HSPH
      HWAK=-HWAK
   47 CONTINUE
      CALL TRAP(TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH)
      ACLSKT=ACLSKT+SKT
      ACLWAK=ACLWAK+WAK
      ACLSPH=ACLSPH+SPH
      CWINDS=ACLSKT
      CWINDW=ACLWAK
      CWINDS=ACLSPH
      CALL TRNCHK(ACLSKT,ACLWAK,ACLSPH)
      IF(TEST)GO TO 998
   48 CONTINUE
      IF(TIME.GT.TTR)GO TO 50
      CALL TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)
      IF(TEST)GO TO 998
   50 CONTINUE
      ACLC=(ACLWAK+ACLSPH)*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL))
      ACLS=ACLSKT*OWF(NWL,NSOIL)
C
C     COMPUTE THE TRANSMITTANCE ALONG THE LINE OF SIGHT
C
      TRNLOS=EXP(-ACLS-ACLC)
      GO TO 999
  998 TRNLOS=0.0
  999 RETURN
      END
```

```
      SUBROUTINE TRNCHK(ACLS,ACLW,ACLC)
C
C   THIS IS A SUBROUTINE TO CHECK IF THE CONCENTRATIONS ARE HIGH ENOUGH
C   SO THAT THE TRANSMITTANCE WILL BE BELOW A GIVEN LEVEL.
C
C   INPUTS
C
C      ACLS  - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT
C              DUE TO THE SKIRT
C
C      ACLW  - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT
C              DUE TO THE WAKE
C
C      ACLC  - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT
C              DUE TO THE BUOYANT SPHERE
C
C      ALL OTHER NECESSARY INFORMATION IS PASSED VIA COMMON BLOCKS
C
C   OUTPUT
C
C      TEST - LOGIICAL VARIABLE PASSED IN COMMON/TRANNY/ THAT IS .TRUE.
C             IF THE CONCENTRATION IS SUCH THAT THE TRANSMITTANCE
C             ALONG THE LINE OF SIGHT WILL BE LESS THAN THE THRESHOLD
C             VALUE AND IS .FALSE. OTHERWISE
C
C
C   FUNCTIONS ANT SUBROUTINE NEEDED
C
C      NONE
C*****************************************************************
C
      DIMENSION OWF(5,2),OWFC(5)
      LOGICAL TEST
      COMMON/CARB/RCARB1,RCARB2
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
      DATA OWFC/1.,.95,.5,.2,1.E-03/
      TEST=.FALSE.
      ACL=(ACLW+ACLC)*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL))
      ACL=ACL+ACLS*OWF(NWL,NSOIL)
      IF(ACL.GT.THRESH)TEST=.TRUE.
  999 RETURN
      END
```

```
      SUBROUTINE TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)
C
C  ROUTINE FOR DETERMINING CONTRIBUTION FROM SPHERE AND WAKE
C  BEFORE THE BUOYANT SPHERE HAS CONVERTED TO THE WIND MODEL.
C
C  INPUTS
C
C     XNORM  -DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER
C
C     TIME   -TIME AT WHICH TRANSMITTANCE IS DESIRED
C
C  OUTPUTS
C
C     ACLWAK -CONTRIBUTION FROM WAKE TO OPTICALLY WEIGHTED CONCENTRATION
C             ALONG GIVEN LINE OF SIGHT
C
C     ACLSPH -CONTRIBUTION FROM BUOYANT SPHERE TO OPTICALLY WEIGHTED
C             CONCENTRATION ALONG GIVEN LINE OF SIGHT
C
C  SUBROUTINES NEEDED
C
C     AVRG   -COMPUTE THE AVERAGE OF THE MOMENTS FOR THE DISCS
C
C     WIN    -COMPUTE THE WIND SPEED AT A GIVEN HEIGHT
C
C     AMOUNT-COMPUTE THE DISTRIBUTION OF THE LOADING BETWEEN THE BUOYANT
C            SPHERE AND WAKE.
C
C     CONLEN-COMPUTE THE LENGTH OF INTERSECTION OF NON-HORIZONTAL LINE
C            OF SIGHT WITH A CONICAL SHAPED WAKE
C
C     TRNCHK-ROUTINE TO CHECK IF THE OBSCURATION IS SUCH THAT THE
C            TRANSMITTANCE IS LESS THAN A USET SPECIFIED AMOUNT
C
C*********************************************************************
      LOGICAL HORIZ,TEST
      DIMENSION CENTER(3)
      COMMON/MODE/HORIZ
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON/BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM
      COMMON/PRTINF/R0,VGRAV(3),NPRTS
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
      COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
     +           DMM(600),DMMY(401)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/LOS/TR(3),RE(3),U(3)
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
      COMMON/ACL/CWINDS,CWINDC,CWINDW
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      ACLWAK=0.0
      ACLSPH=0.0
C
C  DETERMINE THE RADIUS OF THE BASE OF THE CONE
C
      ZX=5.0
      CALL AVRG(ZX,TIME,QTOT,XBAVG,SIG2X,SIG2Y)
      IF(QTOT.LT.1.E-10)GO TO 33
      SIGX=SQRT(SIG2X)
      SIGY=SQRT(SIG2Y)
      RB=SQRT(SIGX*SIGY)
      GO TO 35
   33 CALL WIN(5.0,UW,V)
      XBAVG=UW*TIME
      RB=0.0
   35 VOLWAK=(PI/3.)*(ZCM-5.0)*(RSPH**2+RSPH*RB+RB**2)
      VOLSPH=(4./3.)*PI*RSPH**3
      CALL AMOUNT(VOLSPH,WAKAL,SPHAL)
      DO 80 J=1,ITOT
      IF (VOLWAK.LE.0.0) GO TO 68
      XB=XBAVG+DIFF(1,J)
```

```
      YB=DIFF(2,J)
      XC=XCM+DIFF(1,J)
      YC=YCM+DIFF(2,J)
      IF(ABS(U(3)).LE..01)GO TO 110
      ZBOT=5.0
      CALL CONLEN(U,TR,ZCM,ZBOT,XC,YC,RSPH,RB,XB,YB,XNORM,PLENWK)
      ACLW=PLENWK*WAKAL/VOLWAK
      GO TO 69
   68 ACLW=0.0
   69 CONTINUE
      ACLWAK=ACLWAK+ACLW
      CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)
      IF(TEST)GO TO 999
C
C  DETERMINE CONTRIBUTION FROM SPHERE FO A SLANT PATH
C
      CENTER(1)=XCM+DIFF(1,J)-TR(1)
      CENTER(2)=YCM+DIFF(2,J)-TR(2)
      CENTER(3)=ZCM-TR(3)
      CLOSE=U(1)*CENTER(1)+U(2)*CENTER(2)+U(3)*CENTER(3)
      CON=CENTER(1)**2+CENTER(2)**2+CENTER(3)**2-RSPH**2
      RADIC=CLOSE**2-CON
      IF(RADIC.LT.0.0)GO TO 75
      PNEAR=CLOSE-SQRT(RADIC)
      PFAR=CLOSE+SQRT(RADIC)
      PLENSP=AMIN1(PFAR,XNORM)-AMAX1(PNEAR,0.0)
      IF(PLENSP.LT.0.0)PLENSP=0.0
      ACLS=PLENSP*SPHAL/VOLSPH
      GO TO 76
   75 ACLS=0.0
   76 ACLSPH=ACLSPH+ACLS
      CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)
      IF(TEST)GO TO 999
      GO TO 80
C
C  COMPUTE CONTRIBUTIONS FOR SPHERE AND WAKE FOR A HORIZONTAL PATH
C
  110 IF(TR(3).GT.ZCM+RSPH)GO TO 999
      XCEN=XCM+DIFF(1,J)
      YCEN=YCM+DIFF(2,J)
      IF(TR(3).LT.ZCM-RSPH)GO TO 130
      RADIUS=SQRT(RSPH**2-(TR(3)-ZCM)**2)
      CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENSP)
      IF(PLENSP.LT.0.0)PLENSP=0.0
      ACLSPH=ACLSPH+PLENSP*SPHAL/VOLSPH
      CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)
      IF(TEST)GO TO 999
  130 IF(TR(3).GT.ZCM)GO TO 999
      IF(TR(3).LE.5.0)GO TO 999
      ZETA=(TR(3)-5.0)/(ZCM-5.0)
      XCEN=ZETA*XCEN+(1.-ZETA)*XB
      YCEN=ZETA*YCEN+(1-ZETA)*YB
      RADIUS=ZETA*RSPH+(1-ZETA)*RB
      CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENWK)
      IF(PLENWK.LT.0.0)PLENWK=0.0
      ACLWAK=ACLWAK+PLENWK*WAKAL/VOLWAK
      CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)
      IF(TEST)GO TO 999
   80 CONTINUE
  999 RETURN
      END
```

```
      SUBROUTINE UNIT(A,B,XNORM)
      DIMENSION A(2),B(2)
C *** B IS THE NORM OF A, AND XNORM IS THE MAGNITUDE
      XNORM=SQRT(A(1)**2+A(2)**2)
      B(1)=A(1)/XNORM
      B(2)=A(2)/XNORM
      RETURN
      END
```

```
      SUBROUTINE VEHCL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,PHI,NSOIL,
     1                 SILT,NWL,TRNCOR,RECCOR,TIME,DHDT,V0,VEHDIR,
     2                 VEHSPD,VEHWID,VEHWHT,VEHTYP,NEWATM,NEWVEH,
     3                 TRNLOS,NERR)
C
C     THIS ROUTINE CONTROLS THE FLOW OF THE CALCULATION FOR THE
C  VEHICLE GENERATED DUST CLOUD.
C
C  INPUTS
C    SEE DRTRAN FOR DETAILS
C
C  OUTPUTS
C
C     TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT
C
C     NERR   - ERROR CODE
C
C
C***********************************************************************
      DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2)
      DIMENSION TRNCOR(3),RECCOR(3),TRN(3),REC(3),TRNFRM(2,2),ORIG(2),
     1          V1(2),V0(2),VDIR(2)
      LOGICAL DHDT,ERR,NEWATM,NEWVEH
      INTEGER VEHTYP
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA RTPI /1.772453/
      THETAX=THWND*PIRAD
      IF(.NOT.NEWATM)GO TO 100
      CALL ATMCAL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,THETAX,DHDT,ERR)
      IF(.NOT.ERR)GO TO 100
      NERR=7
      GO TO 999
  100 CONTINUE
      IF(.NOT.NEWVEH)GO TO 5
      CALL VSRC(VEHSPD,VEHWID,VEHWHT,VEHTYP,NSOIL,SILT)
      CALL PREVEH(NSOIL,NWL)
    5 CONTINUE
C
C  COMPUTE DIRECTION VECTOR FOR THE VEHICLE MOTION FROM USERS INPUT
C
      ANGL=VEHDIR*PIRAD
      V1(1)=VEHSPD*COS(ANGL)
      V1(2)=VEHSPD*SIN(ANGL)
C
C  COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER
C  DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN
C  THE WIND DIRECTION.
C
      TRNFRM(1,1)=COS(THETAX)
      TRNFRM(2,2)=TRNFRM(1,1)
      TRNFRM(1,2)=SIN(THETAX)
      TRNFRM(2,1)=-TRNFRM(1,2)
      ORIG(1)=V0(1)
      ORIG(2)=V0(2)
C
C  COMPUTE NEW COORDINATES BY MULTIPLING BY THE TRANSFORMATION MATRIX
C
      TRN(3)=TRNCOR(3)
      REC(3)=RECCOR(3)
      DO 20 I=1,2
      TRN(I)=0.0
      REC(I)=0.0
      VDIR(I)=0.0
      DO 10 J=1,2
      TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))
      REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))
      VDIR(I)=VDIR(I)+TRNFRM(I,J)*V1(J)
```

```
   10 CONTINUE
   20 CONTINUE
C
C  CALL VEHTRN ROUTINE TO USE COMPUTED QUADRATIC FITS TO CALCULATE
C  A TRANSMITTANCE
C
      CALL VEHTRN(TRN,REC,TIME,VDIR,TRNLOS)
  999 RETURN
      END
```

```
      SUBROUTINE VEHTRN(TRN,REC,TIME,VDIR,TRNLOS)
C
C      THIS ROUTINE PARAMETERIZES THE LINE CONNECTING THE TRANSMITTER
C    AND RECEIVER IN THE LOCAL COORDINATE SYSTEM AND DOES A TRAPEZOIDAL
C    INTEGRATION FROM VEHICLE TIME=0.0 TO VEHICLE TIME=TIME THE
C    TRANSMITTANCE IS DESIRED.
C
C    INPUTS
C
C      TRN    - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATE
C               SYSTEM
C
C      REC    - THE COORDINATES OF THE RECEIVER IN THE LOCAL COORDINATE
C               SYSTEM
C
C      TIME   - THE PRESENT TIME AT WHICH A TRANSMITTANCE IS DESIRED
C
C      NWL    - INTEGER INDEX FOR WAVELENGTH BEING USED
C
C      NSOIL  - INTEGER INDEX FOR SOIL TYPE
C
C      VDIR   - VECTOR INDICATING THE DIRECTION AND SPEED OF THE VEHICLE
C               IN THE LOCAL COODINATE SYSTEM
C
C    OUTPUT
C
C      TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT AT THE INDICATED
C               TIME
C
C    FUNCTIONS AND SUBROUTINES NEEDED
C
C       GRAND EVALUATES THE INTEGRAND
C
C*****************************************************************
      LOGICAL TEST
      DIMENSION TRN(3),REC(3),TR(3),RE(3),VDIR(2),OWF(5,2),U(3)
      COMMON/MO5/DMMMY(604),DMM(600),
     +                ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
     +                RB(3,25),Z2(3,25)
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
      TEST=.FALSE.
C
C    PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER
C
      XNORM=0.0
      DO 10 I=1,3
      RE(I)=REC(I)
      TR(I)=TRN(I)
      U(I)=RE(I)-TR(I)
      XNORM=XNORM+U(I)**2
   10 CONTINUE
      XNORM=SQRT(XNORM)
      U(1)=U(1)/XNORM
      U(2)=U(2)/XNORM
      U(3)=U(3)/XNORM
C
C       INCREMENT VEHICLE TRAVEL TIME AND CALL GRAND TO COMPUTE THE VALUE
C    OF THE INTEGRAND.  IF THE VALUE OF THE VARIABLE TIME IS GREATER
C    THAN THE MAXIMUM TIME THAT HAS BEEN STORED (APPROX. 373 SEC) THEN
C    ANY DUST PRODUCED MORE THAN TIMES(20) SECONDS IS ASSUMED TO HAVE
C    NO EFFECT ON THE TRANSMITTANCE.
C
C
      IF(TIME.LE.TIMES(ICOUNT))GO TO 11
      TINC=TIMES(ICOUNT)/400.
      TSTART=TIME-TIMES(ICOUNT)
      GO TO 15
   11 TINC=TIME/400.
      TSTART=0.0
```

```
 15 SUM=0.0
    DO 50 I=1,401
    TIVEH=TSTART+FLOAT(I-1)*TINC
    CALL GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)
    IF(I.EQ.1.OR.I.EQ.401)GO TO 20
    SUM=SUM+VALUE
    GO TO 40
 20 SUM=SUM+VALUE/2.
 40 CONTINUE
    SUM1=SUM*TINC
    ACLWAK=0.0
    ACLSPH=0.0
    CALL TRNCHK(SUM1,ACLWAK,ACLSPH)
    IF(TEST)GO TO 998
 50 CONTINUE
    SUM=SUM*TINC
    ACL=OWF(NWL,NSOIL)*SUM
    TRNLOS=EXP(-ACL)
    GO TO 999
998 TRNLOS=0.0
999 RETURN
    END
```

```
      SUBROUTINE VSRC(VSPD,VWID,VWHT,VEHTYP,NSOIL,SILT)
C
C   THIS SUBROUTINE INITIALIZES THE DUST CLOUD PRODUCED BY A VEHICLE
C
C   INPUTS
C
C      VSPD  - THE VELOCITY OF THE VEHICLE
C
C      VWID  - WIDTH OF THE VEHICLE
C
C      VWHT  - WEIGHT OF THE VEHICLE IN KGS.
C
C      NSOIL - SOIL TYPE
C
C      SILT  - SILT CONTENT OF THE SOIL
C
C
C   OUTPUTS
C
C    STORED IN COMMON/DISCS/ AND COMMON/PRE/
C
C      NDSCS - NUMBER OF DISCS (FOR A VEHICLE ONLY ONE)
C
C      TDSC -  TIME OF RELEASE OF THE DISCS
C
C      XDSC -  X POSITION OF DISC AT TIME OF RELEASE
C
C      ZDSC -  HEIGHT OF RELEASE OF THE DISC
C
C      R2DSC - SQUARE OF THE RADIUS OF THE DISC
C
C   *****************************************************************
      INTEGER VEHTYP
      COMMON/PRTINF/R0,VGRAV(3),NPRTS
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
      COMMON/PRE/Z,RT2DZ
      COMMON/VL/VLOAD
      NDSCS=1
      NPRTS=1
      VGRAV(1)=0.0
      QDSC(1,1)=1.0
C
C   INITIALIZE THE VEHICLE SOURCE FOR WHEELED VEHICLES
C
      ZDSC(1)=VWID/8.
      Z=VWID/4.
      ZZ=VWID/4.
      DZ=DIFFUS(Z0,ZL,ZZ)
      DX=DXZ0*DZ
      TDSC(1)=-9.*(VWID**2)/512./DZ
      TOF=-TDSC(1)
      CALL MOMENT(VGRAV,Z,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)
      XDSC(1)=-XBAR
      A=1.0
      B=SIGW2+SIGP2
      C=4.*((SIGW2*SIGP2)-((VWID/3.)**4))
      RAD=B**2-A*C
      R2DSC(1)=-B+SQRT(RAD)
      RT2DZ=SQRT(2.*DZ)
C
C   INTIALIZE LOADING FOR VEHICLE (VLOAD IN KG/SEC)
C
   20 SILTPC=100.*SILT
      A=3.8E-9
      IF(VEHTYP.GT.0)A=1.52E-08
      Q=A*VSPD*VWHT*SILTPC
      ALPHA=240.
```

```
      VLOAD=ALPHA*VSPD*Q                                                VRS00710
  999 RETURN                                                            VRS00720
      END                                                               VRS00730
```

```
      SUBROUTINE VSUM(A,B,S,C)
      DIMENSION A(2),B(2),C(2)
C *** C=A+S*B WHERE A,B,C ARE VECTORS AND S IS SCALAR
      DO 14 J=1,2
   14 C(J)=A(J)+S*B(J)
      RETURN
      END
```

```
      SUBROUTINE WIN(Z,U,V)
      COMMON/STARS/USTAR,TSTAR,ZSTAR
      COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
      COMMON/BUOYCL/Y(8),SPHNS(3),RISTIM
C *************************************************************
C
C  PURPOSE
C
C       TO COMPUTE THE WIND SPEED AT A SPECIFIED HEIGHT
C
C       INPUTS
C
C         Z    HEIGHT AT WHICH WIND SPEEDS ARE DESIRED
C
C       OUTPUTS
C
C         U    WIND SPEED IN THE DIRECTION OF THE GROUND WIND
C
C         V    WIND SPEED PERPENDICULAR TO THE GROUND WIND
C
C       CALLED BY DIFEQ
C
C       SUBROUTINES AND FUNCTIONS NEEDED
C
C          WNDCAL    CALCULATES SCALED WIND SPEED
C
C*************************************************************
      IF(Z.GT.ZSTAR)GO TO 100
      U=USTAR*WNDCAL(Z0,ZL,Z)
      V=0.0
      GO TO 999
  100 UE=C*EXP(-ALP*Z)*COS(ALP*Z)-PYF
      VE=-C*EXP(-ALP*Z)*SIN(ALP*Z)+PXF
      U=UHAT*UE+VHAT*VE
      V=-VHAT*UE+UHAT*VE
  999 RETURN
      END
```

```
      FUNCTION WNDCAL(Z0,ZL,Z)
C ******************************************************************
C
C
C
C     PURPOSE
C
C     TO CALCULATE THE WIND SPEED, U/U*, SCALED BY THE FRICTION
C     VELOCITY FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV LENGTH AT A
C     SPECIFIED HEIGHT.
C
C
C     INPUTS
C
C     Z0       THE FRICTION HEIGHT IN METERS.
C     ZL       THE MONIN-OBUKHOV LENGTH IN METERS.
C     Z        THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED
C              IN METERS
C
C
C     RETURNS VELOCITY SCALED BY FRICTION VELOCITY
C
C
C     CALLED BY ATMCAL,WIN AND RISE
C
C ******************************************************************
      LOGICAL LOW
      COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      PSIM(Z,Z1)=ALOG((1.-Z)/(1.-Z1))-ALOG((1.+Z)/(1.+Z1))+
     $2.*(ATAN(Z)-ATAN(Z1))
      PSIMS(Z)=-7.*Z
      PHIM(Z)=(1.-16.*Z)**(-.25)
C
C     PHIM       THE SHEAR OF MOMENTUM
C     PSIM       THE UNIVERSAL FUNCTION FOR THE DEVIATION FROM
C                LOGARITHMIC WIND VELOCITY BOUNDARY LAYER PROFILE IN AN
C                UNSTABLE ATMOSPHERE
C     PSIMS      THE SAME AS PSIM FOR A STABLE ATMOSPHERE
C
      IF(ABS(ZL).LE.1.E3)GO TO 100
      WNDCAL=ALOG(1.+Z/Z0)
      GO TO 999
  100 CONTINUE
      P=SIGN(1.,ZL)
      LOW=.TRUE.
      S=Z/ZL
      IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10
      S=AMIN1(S,1.5)
      S=AMAX1(S,-2.)
      LOW=.FALSE.
   10 CONTINUE
      IF(P)120,130,130
  120 S=1./PHIM(S)
      S1=Z0/ZL
      S0=1./PHIM(S1)
      WNDCAL=PSIM(S,S0)
C
C DETERMINE THE CONSTANTS FOR MATCHING AT Z/ZL=-2.
C
      S2=-2.
      AW=-3.*((1-16.*(S2))**(-.25))*((-S2)**(1./3.))
      CW=-1.*AW*(-S2)**(-1./3.)
      GO TO 52
  130 CONTINUE
      PSI=PSIMS(S)
      WNDCAL=ALOG(1.+S*ZL/Z0)-PSI
C
C  FIND THE CONSTANTS FOR MATCHING OF STABLE PROFILE AT Z/ZL=1.5
C
```

```
      S2=1.5
      BW=1./(Z0/ZL+S2)+7.
      DW=-1.*S2*BW
   52 CONTINUE
      IF(LOW)GO TO 999
      IF(P)53,53,54
   53 WNDCAL=WNDCAL+CW+AW*(-ZL/Z)**(1./3.)
      GO TO 999
   54 WNDCAL=WNDCAL+DW+BW*Z/ZL
  999 WNDCAL=WNDCAL/.4
  998 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NMMW(FREQGH,ICLMAT,MMTRAN,IERR)
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
     +           FOGPRB,WNDVEL,WNDDIR,IPASCT
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /GEOMET/PTS(15),IGEOSW
C-----------------------------------------------------------------------
C    PURPOSE: TO CALCULATE THE EXTINCTION AND BACKSCATTER AT MILLIMETER
C              FREQUENCIES 10 TO 1000 GHZ, DUE TO
C              GASEOUS ABSORPTION, FOG(ICE AND WATER) AND
C              CLOUD BULK ATTENUATION, RAIN, SNOW EXTINCTION.
C-----------------------------------------------------------------------
C
C** INPUT TO THE NMMW MODULE IS PERFORMED THROUGH A CARD ORDER-
C** INDEPENDENT INPUT TECHNIQUE. A FOUR-LETTER IDENTIFIER IN COLS.
C** 1-4 OF EACH RECORD SPECIFIES THE TYPE OF DATA BEING READ BY THE
C** MODULE. THE INPUT CARDS MAY APPEAR IN ANY ORDER WITH THE EXCEPTION
C** OF OF THE (GO) END OF READ SENTINEL, WHICH MUST BE THE LAST CARD
C** READ. ALL OF THE FOLLOWING CARDS ARE READ IN UNDER THE FORMAT
C** (A4,1X,3(E10.4,1X)) :
C
C-----------------------------------------------------------------------
C       CARD IDENTIFIER : PATH
C       VARIABLES READ  : MMWPTH       (PATH LENGTH (KM))
C-----------------------------------------------------------------------
C       CARD IDENTIFIER : ATMO
C       VARIABLES READ  : TEMP1        (TEMPERATURE (DEG C))
C                         PRESS1       (PRESSURE (MB))
C                         ABSHUM       (IF .GT. 0. ABSOLUTE
C                                       HUMIDITY (GM/M**3))
C                                      (IF .LT. 0. RELATIVE
C                                       HUMIDITY (%))
C   ***NOTE: TEMP1,PRESS1,ABSHUM WILL BE PASSED FROM CLIMAT IF
C            ICLMAT=1. IN THAT EVENT, THE (ATMO) CARD IS NOT NEEDED.
C-----------------------------------------------------------------------
C       CARD IDENTIFIER : FOGD
C       VARIABLES READ  : FOGDEN       (FOG DENSITY (LIQUID WATER,
C                                      GM/M**3))
C-----------------------------------------------------------------------
C       CARD IDENTIFIER : RAIN
C       VARIABLES READ  : RAINRT       (RAIN RATE (MM/HR))
C-----------------------------------------------------------------------
C       CARD IDENTIFIER : SNOW
C       VARIABLES READ  : SNOWRT       (SNOW RATE (MM/HR) WATER EQUIV.)
C-----------------------------------------------------------------------
C       CARD IDENTIFIER : GO
C       VARIABLES READ  : NONE         (END OF READ SENTINEL)
C-----------------------------------------------------------------------
C
C  MAIN ROUTINE CALLS NMMW. NMMW THEN CALLS
C     (1)MMWGS(GAS ABSORPTION), (2)MMWFG(FOG ABSORPTION,BACKSCATTER),
C     (3)MMRAN(RAIN EXTINCTION,BACKSCATTER),
C     (4)MMSNO(SNOW EXTINCTION,BACKSCATTER), AND RETURNS TO MAIN NMMW
C
C  LOCAL VARIABLES
C-----------------------------------------------------------------------
      REAL MMWPTH,MMTRAN,MMBSXS
      DIMENSION DAT(3),IAL(6)
      DATA IAL/2HPA  ,2HAT  ,2HFO  ,2HRA  ,2HSN  ,2HGO  /
      IERR=0
      GASABS=0.
      FOGEXT=0.
      RAINEX=0.
      FOGBS=0.
      RAINBS=0.
      SNOWBS=0.
    5 READ(IOIN,400)IALFA,IALFA2,(DAT(L),L=1,3)
  400 FORMAT(2A2,1X,3(E10.4,1X))
      IF(IALFA.EQ.IAL(1)) GO TO 10
      IF(IALFA.EQ.IAL(2)) GO TO 20
```

```
      IF(IALFA.EQ.IAL(3)) GO TO 30
      IF(IALFA.EQ.IAL(4)) GO TO 40
      IF(IALFA.EQ.IAL(5)) GO TO 50
      IF(IALFA.EQ.IAL(6)) GO TO 60
      WRITE(IOOUT,450)IALFA,IALFA2
  450 FORMAT(1H0,20X,2A2,22H IS AN INCORRECT LABEL //)
      GO TO 300
   10 MMWPTH=DAT(1)
      GO TO 5
   20 TEMP1=DAT(1)
      PRESS1=DAT(2)
      ABSHUM=DAT(3)
      GO TO 5
   30 FOGDEN=DAT(1)
      GO TO 5
   40 RAINRT=DAT(1)
      GO TO 5
   50 SNOWRT=DAT(1)
      GO TO 5
   60 CONTINUE
      IF(IGEOSW.NE.1)GO TO 99
      MMWPTH=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+
     +(PTS(6)-PTS(3))**2)
   99 CONTINUE
      IF(ICLMAT.EQ.1) TEMP1=TEMP
      IF(ICLMAT.EQ.1) PRESS1=PRESS
      IF(ICLMAT.EQ.1) ABSHUM=AH
      WRITE(IOOUT,500) TEMP1,PRESS1,ABSHUM,FOGDEN
      WRITE(IOOUT,600) RAINRT,SNOWRT,FREQGH,MMWPTH
C CHECK MODEL INPUTS FOR RANGE OF VALIDITY.
      IF(PRESS1.GE.500.) GO TO 100
      IERR=1
      WRITE(IOOUT,800)
  100 IF(FREQGH.LE.1000..AND.FREQGH.GE.10.) GO TO 150
      IERR=1
      WRITE(IOOUT,900)
  150 IF(IERR.EQ.1) GO TO 300
C CHANGE UNITS
      PRSTOR=PRESS1/TORRMB
      TEMPDK=TEMP1+CDEGK
C CALL INDIVIDUAL MODULES FOR GAS, FOG/CLOUD, SNOW, RAIN EXTINCTION
      CALL MMWGS(TEMPDK,PRSTOR,ABSHUM,FREQGH,GASABS)
      IF(FOGDEN.GT.1.E-10)
     +    CALL MMWFG(FOGDEN,TEMPDK,FREQGH,FOGEXT,FOGBS)
      IF(RAINRT.GT.1.E-10)
     +    CALL MMRAN(RAINRT,TEMPDK,FREQGH,2.,RAINEX,RAINBS)
      IF(SNOWRT.GT.1.E-10)
     +    CALL MMSNO(SNOWRT,TEMPDK,FREQGH,SNOWEX,SNOWBS)
C COMPUTE TRANSMISSION
      TOTEXT=GASABS+FOGEXT+RAINEX+SNOWEX
      MMBSXS=FOGBS+RAINBS+SNOWBS
      MMTRAN=EXP(-MMWPTH*TOTEXT)
C CHANGE UNITS FOR ABSORPTION/EXTINCTION FROM 1/KM TO DB/KM.
      DBKM=4.343
      GASABS=GASABS*DBKM
      FOGEXT=FOGEXT*DBKM
      RAINEX=RAINEX*DBKM
      SNOWEX=SNOWEX*DBKM
      WRITE(IOOUT,700) GASABS,FOGEXT,RAINEX,SNOWEX,MMTRAN
      WRITE(IOOUT,750) FOGBS,RAINBS,SNOWBS,MMBSXS
C  COMPUTATION COMPLETED
  300 RETURN
C
500   FORMAT(1H0,///,47X,12HTEMPERATURE ,14X,F8.3,
     +       10H DEGREES C,/,47X,9HPRESSURE ,17X,F8.3,
     +       3H MB,/,47X,17HABSOLUTE HUMIDITY,9X,F8.3,
     +       7H G/M**3/,47X,11HFOG DENSITY,15X,F8.3,
     +       7H G/M**3)
600   FORMAT(1H ,46X,9HRAIN RATE,17X,F8.3,6H MM/HR,/
     +       ,47X,9HSNOW RATE,17X,F8.3,6H MM/HR,/
```

```
     +        ,47X,9HFREQUENCY,17X,F8.3,4H GHZ/
     +        ,47X,11HPATH LENGTH,15X,F8.3,3H KM)
 700  FORMAT(1H0,46X,14HGAS ABSORPTION,10X,E10.4,6H DB/KM,/
     +        ,47X,15HFOG EXTINCTION ,9X,E10.4,6H DB/KM,/
     +        ,47X,16HRAIN EXTINCTION ,8X,E10.4,6H DB/KM,/
     +        ,47X,16HSNOW EXTINCTION ,8X,E10.4,6H DB/KM,/
     +        ,47X,13HTRANSMISSION ,11X,E10.4/)
  750 FORMAT(1H ,46X,15HFOG BACKSCATTER,9X,E10.4,10H M**2/M**3,/
     +        ,47X,16HRAIN BACKSCATTER,8X,E10.4,10H M**2/M**3,/
     +        ,47X,16HSNOW BACKSCATTER,8X,E10.4,10H M**2/M**3,/
     +        ,47X,17HTOTAL BACKSCATTER,7X,E10.4,10H M**2/M**3)
 800  FORMAT(1H0,47X,41HPRESSURE LESS THAN 500 MB, GAS ABSORPTION,
     +        19H WILL BE INACCURATE)
 900  FORMAT(1H0,47X,31HFREQUENCY<10. GHZ, OR >1000 GHZ,
     +        42H CALULATION WILL FAIL, USE OTHER FREQUENCY)
      END
```

```
      FUNCTION AB(WA,A,CE,B,C)
      AB=A*EXP(-ABS((ALOG10(1.E4*WA/CE)/B))**C)
      RETURN
      END
```

```
FUNCTION DOP(WA,A,CE1,B,C,CE2,D,E,CE3,F,G)
V=1./WA
V2=V*V
H1=CE1*CE1-V2
H2=CE2*CE2-V2
H3=CE3*CE3-V2
DOP=SQRT(A+B*H1/(H1*H1+C*V2)+D*H2/(H2*H2+E*V2)+F*H3/(H3*H3+G*V2))
RETURN
END
```

```
      SUBROUTINE INTRP(A,B,T,F,TT,FF,AA,BB,J)
C
C-----------------------------------------------------------------------
C     PURPOSE: TO DO FREQUENCY AND TEMPERATURE INTERPOLATION
C-----------------------------------------------------------------------
      DIMENSION A(9,3), B(9,3), F(9), T(3)
C
      IF(TT.LT.T(1)) TT=T(1)
      DO 11 J=2,3
         IF(TT.LT.T(J)) GO TO 14
   11 CONTINUE
      TT=T(3)
      J=3
   14 CONTINUE
      DO 15 I=2,9
      IF(FF.LT.F(I)) GO TO 16
   15 CONTINUE
      FF=F(9)
      I=9
   16 FF=ALOG10(FF)
      F0=ALOG10(F(I))
      F1=ALOG10(F(I-1))
      FF0=(F0-FF)/(F0-F1)
      FF1=(FF-F1)/(F0-F1)
      TF0=(T(J)-TT)/(T(J)-T(J-1))
      TF1=(TT-T(J-1))/(T(J)-T(J-1))
      A11=ALOG10(A(I-1,J-1))
      A01=ALOG10(A(I,J-1))
      A10=ALOG10(A(I-1,J))
      A00=ALOG10(A(I,J))
      APJ1=A11*FF0+A01*FF1
      APJ0=A10*FF0+A00*FF1
      AA=APJ1*TF0+APJ0*TF1
      B11=ALOG10(B(I-1,J-1))
      B10=ALOG10(B(I-1,J))
      B01=ALOG10(B(I,J-1))
      B00=ALOG10(B(I,J))
      BPJ1=B11*FF0+B01*FF1
      BPJ0=B10*FF0+B00*FF1
      BB=BPJ1*TF0+BPJ0*TF1
      AA=10.**AA
      BB=10.**BB
      RETURN
      END
```

```
      SUBROUTINE MMH2O(V,T,PTOT,PH2O,DATH2O,ABH2O)
C
C*******************************************************************
C
C   ROUTINE TO CALCULATE H2O VAPOR ABSORPTION FOR 0 TO 350 GHZ.
C
C   INPUTS ARE: WAVENUMBER(/CM), TEMPERATURE(KELVIN), TOTAL
C     PRESSURE(TORR), H2O VAPOR PRESSURE(TORR), LINE DATA ARRAY.
C   OUTPUT IS: H2O VAPOR ABSORPTION
C
C   CALLED BY MMWGS   MAKES NO CALLS
C
C   LOCAL VARIABLES:
C     WCD        VAPOR COLUMN DENSITY(/CM/CM/KM)
C     CT         LINE STRENGTH TEMPERATURE CORRECTION
C     CA         LINE WIDTH SELF BROADENING AND TEMP. CORRECTION
C     SA         CORRECTED LINE STRENGTH
C     GA         CORRECTED LINE WIDTH
C     ABS        SINGLE LINE ABSORPTION(/KM)
C*************************************************************************
      DIMENSION DATH2O(37,4)
C
      ABH2O=0.
      WCD=7.33994E26*PH2O/760./T*(PFR(T))
      CT=4.860773E-3*(T-296.)/T
      CA=((296./T)**.62)*(PTOT+(4.*PH2O))/760.
C
      DO 500 L=1,37
         SA=DATH2O(L,2)*WCD*EXP(DATH2O(L,4)*CT)
         GA=DATH2O(L,3)*CA
         ABS=SA*SUPK(V,DATH2O(L,1),GA)
         ABH2O=ABH2O+ABS
500   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MMIDX(XL,T,ICE,H2OAB,H2OK2)
C   REF: RAY, APPLIED OPTICS, VOL. 11, P. 1836,(1972)
      COMPLEX CINDX,XK
      XXL=XL/10.
      TT=T-273.16
      T3=TT-25.
      IF(ICE.NE.0) GO TO 150
C  PARAMETERS FOR WATER
100   EFIN=5.27137+.0216474*TT-1.31198E-3*TT*TT
      ALFA=-16.8129/T+.0609265
      XLS=3.3836E-4*EXP(2513.98/T)
      SIGMA=1.25664E9
      ES=78.54*(1.-4.579E-3*T3+1.19E-5*T3*T3-2.8E-8*T3*T3*T3)
      GO TO 200
C
C     PARAMETERS FOR ICE
  150 EFIN=3.168
      ALFA=0.288+0.0052*TT+2.3E-4*TT*TT
      XLS=9.990288E-5*EXP(1.32E4/(1.9869*T))
      SIGMA=1.26*EXP(-1.25E4/(1.9869*T))
      ES=203.168+2.5*TT+0.15*TT*TT
200   U=(ES-EFIN)*(XLS/XXL)**(1.-ALFA)
      Y=1.+2.*(XLS/XXL)**(1.-ALFA)*SIN(ALFA*1.57079633)+
     1    (XLS/XXL)**(2-2*ALFA)
      EP=EFIN+((ES-EFIN)+U*SIN(ALFA*1.57079633))/Y
      EPP=(U*COS(ALFA*1.57079633))/Y+SIGMA*XXL/1.88496E11
      RE=SQRT((EP+SQRT(EP*EP+EPP*EPP))/2.)
      AI=-EPP/2./RE
      IF(ICE.NE.0) GO TO 400
C
      IF(XXL.LE..034) GO TO 307
      IF(XXL.GT..1)  GO TO 311
306   R2=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,
     +          259913.,161.29,43319.7,27661.2)
      R2=R2+R2*T3*1.E-3*EXP((2.5E-5*XXL)**.25)
      RE=RE*(XXL-.034)/.066+R2*(.1-XXL)/.066
      GO TO 311
307   RE=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,
     +          259913.,161.29,43319.7,27661.2)
      RE=RE+RE*T3*1.E-3*EXP((2.5E-5*XXL)**.25)
311   CONTINUE
      IF(XXL.GT. .3) GO TO 500
      AI=AI+AB(XXL,.25,300.,.47,3.)+AB(XXL,.39,17.,.45,1.3)
     +      +AB(XXL,.41,62.,.35,1.7)
      GO TO 500
C
400   CONTINUE
      IF(XXL.GT.0.08) GO TO 500
  405 RC=DOP(XXL,1.225,1652.9,1.12082E6,46E-11,909.09,416441.,118852.,
     +          223.2,47031.8,126834.)
      RE=RE*(XXL-0.02)/0.06+R2*(0.08-XXL)/0.06
      AI=AI+AB(XXL,.242,62.,.23,1.6)+AB(XXL,.581,44.8,0.055,1.)
C
500   CINDX=CMPLX(RE,AI)
      XK=(CINDX*CINDX-1)/(CINDX*CINDX+2)
      H2OAB=AIMAG(-XK)
      H2OK2=XK*CONJG(XK)
      RETURN
      END
```

```
      SUBROUTINE MMOXY(V,T,PTOT,PH2O,DATAO2,ABSO2)
C
C*****************************************************
C   ROUTINE TO CALCULATE ABSORPTION DUE TO OXYGEN. METHOD IS THAT OF
C       LIEBE, GIMMESTAD,& HOPPONEN. IEEE TRANS. ANT. PROP. V.25, P327.
C
C   INPUTS ARE:  FREQUENCY(GHZ), TEMPERATURE(KELVIN), TOTAL
C       PRESSURE(TORR), H2O VAPOR PRESSURE(TORR), O2 LINE DATA
C       ARRAY.
C   OUTPUTS ARE: O2 ABSORPTION (1/KM)
C
C   CALLED FROM MMWGS  CALLS NO OTHER ROUTINES.
C
C   LOCAL VARIABLES:
C       T2          300./T
C       PHI         TEMPERATURE CORRECTION FOR LINE STRENGTHS
C       S           CORRECTED LINE STRENGTH(HZ TORR)
C       GAMMA       CORRECTED LINE WIDTH (1/GHZ.)
C       XIF         LINE INTERFERENCE FACTOR
C       VMI         DATAO2(L,1)-V
C       VPL         DATAO2(L,1)+V
C       PROFIL      MODIFIED VANVLECK-WEISSKOPF LINE SHAPE
C
      DIMENSION DATAO2(42,6)
C
      ABSO2=0.
      T2=300./T
C
      DO 500 L=1,42
         K=IFIX(DATAO2(L,6))
         PHI=T2*T2*T2*EXP(-6.895E-3*K*(K+1)*(T2-1.))
         S=0.2095*PTOT*DATAO2(L,2)*PHI
         GAMMA=DATAO2(L,3)*(.929*PTOT*T2**.9+1.3*T2*PH2O)*1.E-3
         XIF=DATAO2(L,4)*T2**DATAO2(L,5)*PTOT*1.E-3
         VMI=DATAO2(L,1)-V
         VPL=VMI+2*V
         PROFIL=(V/DATAO2(L,1))*((GAMMA-VMI*XIF)/(VMI*VMI+GAMMA*GAMMA)+
     *          (GAMMA-VPL*XIF)/(VPL*VPL+GAMMA*GAMMA))
C
         ABSO2=ABSO2+S*PROFIL*V*4.192E-5
C
500   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MMRAN(RAINRT,T,FREQ,RTYPE,GRAIN,BSRAIN)
C   REF: OLSEN, ET.AL., IEEE ANT. PROP., VOL.26, P.318(1978)
C-----------------------------------------------------------------
C   ROUTINE TO COMPUTE THE ATTENUATION DUE TO RAIN, FOR
C      FREQUENCIES BETWEEN 10 & 1000 GHZ; BACKSCATTER ALSO.
C
C   INPUTS ARE: FREGUENCY(GHZ), TEMPERATURE(KELVIN), RAIN RATE(MM/HR)
C   OUTPUT IS ATTENUATION (1/KM), BACKSCATTER (MM**2/MM**3)
C
C   CALLED FROM MMWMOD  CALLS NO SUBROUTINES.
C
C   LOCAL VARIABLES:
C       ALFA,BETA                     A AND B PARAMETERS FUNCTION OF FREQ,
C                                      TEMPERATURE, AND RAIN TYPE
C       BSAT,BSBT                     POLYNOMIAL COEF.S FOR LINEAR FIT
C                                      TO BACKSCATTER= ALF*RAINRT**BET
C-----------------------------------------------------------------
      DIMENSION ALFA(9,3,3), BETA(9,3,3), F(9), TK(3),A(9,3),B(9,3),
     +          BSAT(6,2), BSBT(6,2)
      DATA ALFA/1.42E-2,3.34E-2,.197,.404,1.11,1.76,2.36,2.72,2.89,
     D          1.14E-2,2.82E-2,.180,.387,1.18,1.89,2.46,2.78,2.98,
     D          7.52E-3,2.20E-2,.167,.368,1.24,2.05,2.58,2.84,2.89,
     M          1.59E-2,3.94E-2,.275,.579,1.35,1.80,2.07,2.22,2.00,
     M          1.36E-2,3.68E-2,.268,.579,1.42,1.88,2.10,2.24,2.00,
     M          1.01E-2,3.57E-2,.269,.572,1.46,1.96,2.13,2.24,2.01,
     T          1.51E-2,4.20E-2,.376,.619,.871,1.05,1.02,.972,.857,
     T          1.69E-2,4.66E-2,.372,.629,.909,1.04,.997,.976,.856,
     T          1.90E-2,5.89E-2,.360,.610,.936,1.02,.992,.997,.856/,
     +     BETA/.932,.954,1.016,1.027,.943,.846,.766,.704,.613,
     D          .968,1.003,1.053,1.053,.941,.829,.749,.692,.612,
     D          1.024,1.106,1.107,1.082,.936,.807,.725,.679,.613,
     M          1.094,1.088,1.002,.904,.753,.686,.645,.618,.611,
     M          1.150,1.118,1.007,.905,.742,.677,.641,.614,.610,
     M          1.260,1.160,.999,.900,.732,.667,.635,.612,.610,
     T          1.087,1.027,.784,.714,.657,.610,.608,.612,.614,
     T          1.076,1.010,.783,.709,.650,.610,.613,.611,.615,
     T          1.079,.966,.782,.708,.642,.615,.615,.607,.615/
      DATA TK/263.16,273.16,293.16/,
     +     F/10.,15.,35.,50.,95.,140.,225.,310.,1000./
      DATA BSAT/-.8824881E+01,-.1029998E-01,+.2451205E-04,
     +        -.2462900E-07,+.6507628E-10,-.1856080E-12,
     +        -.2127020E+02,+.6906017E+00,-.1924260E-01,
     +        +.3035233E-03,-.2545323E-05,+.8673581E-08/,
     + BSBT/+.7901887E+00,-.1900189E-02,+.6341350E-05,
     +        +.3186429E-08,-.5933950E-10,+.9056715E-13,
     +        +.1361993E+01,+.3628100E-01,-.2461284E-02,
     +        +.4805257E-04,-.3988064E-06,+.1193458E-08/
C
      TR=T
      GRAIN=0.
      BSRAIN=0.
      IF(FREQ.LT.10.) GO TO 200
      FR=FREQ
C
      ITYPE=IFIX(RTYPE+0.1)
      DO 10 I=1,3
        DO 10 J=1,9
          A(J,I)=ALFA(J,I,ITYPE)
          B(J,I)=BETA(J,I,ITYPE)
   10 CONTINUE
C
      CALL INTRP(A,B,TK,F,TR,FR,AA,BB,J)
      GRAIN=(AA*RAINRT**BB)/4.343
C
C     CALCULATIONS FOR RAIN BACKSCATTER.
C
      AA=0.
      BB=0.
      IA=2
      IB=2
```

```
      IF(FREQ.GT.87.) IA=1
      IF(FREQ.GT.82.) IB=1
      DO 100 K=1,6
      AA=AA+BSAT(K,IA)*(FREQ**(K-1))
      BB=BB+BSBT(K,IB)*(FREQ**(K-1))
100   CONTINUE
      AA=EXP(AA)
      BSRAIN=AA*RAINRT**BB
 200  RETURN
      END
```

```
      SUBROUTINE MMSNO(SNRT,TK,FQ,SNEX,SNBS)
C
C-----------------------------------------------------------------------
C     PURPOSE: TO COMPUTE SNOW EXTINCTION AND BACKSCATTER X-SECTION.
C-----------------------------------------------------------------------
C     INPUTS: SNOWFALL RATE(MM/HR), TEMPERATURE(KELVIN), FREQUENCY(GHZ)
C
C     OUTPUT: SNOW EXTINCTION (1/KM), BACKSCATTER(M**2/M**3).
C
C     CALLED FROM MMMW.
C
C     LOCAL VARIABLES:
C       FR,TS          LOCAL FREQUENCY,TEMPERATURE
C       ITYPE          INTEGER SNOW TYPE
C       XLMDA          WAVELENGTH(MM)
C       FF0,1          FREQUENCY FITTING FACTORS
C       TF0,1          TEMPERATURE FITTING FACTORS
C       AP J0,1        INTERMEDIATE A VALUES
C       BP J0,1        INTERMEDIATE B VALUES
C       AA,BB          TERMS IN EXT(SNOW)=AA*SNOWRATE**BB
C       AABSCT           TERM IN BSCAT(SNOW)=AA*SNOWRATE**1.8
C
C-----------------------------------------------------------------------
      DIMENSION A(9,3),B(9,3),F(9),T(3),BSAT(6,2),BSBT(6,2),SFCT(3)
      DATA A/1.30E-3,2.75E-3,1.25E-2,2.50E-2,8.00E-2,1.65E-1,
     1      3.60E-1,5.91E-1,1.68E+0,
     2      2.07E-2,4.34E-2,1.60E-1,2.00E-1,3.10E-1,4.00E-1,
     3      5.81E-1,6.50E-1,1.11E+0,
     4      6.22E-2,9.67E-2,2.35E-1,3.41E-1,6.10E-1,8.52E-1,
     5      7.83E-1,7.37E-1,5.76E-1/
      DATA B/1.3,1.46,1.6,1.54,1.26,1.1,.89,.79,.6,
     1      1.3,1.2,.95,.80,.75,.67,.65,.64,.60,
     2      1.3,1.2,.95,.80,.75,.67,.65,.64,.60/
      DATA F/10.,15.,35.,50.,95.,140.,225.,312.,1000./
      DATA T/271.,273.,275./,SFCT/1.,3.,4./
      DATA BSAT/-.8824881E+01,-.1029998E-01,+.2451205E-04,
     +      -.2462900E-07,+.6507628E-10,-.1856080E-12,
     +      -.2127020E+02,+.6906017E+00,-.1924260E-01,
     +      +.3035233E-03,-.2545323E-05,+.8673581E-08/,
     + BSBT/+.7901887E+00,-.1900189E-02,+.6341350E-05,
     +      +.3186429E-08,-.5933950E-10,+.9056715E-13,
     +      +.1361993E+01,+.3628100E-01,-.2461284E-02,
     +      +.4805257E-04,-.3988064E-06,+.1193458E-08/
C
      TS=TK
      FS=FQ
      ICE=1
      SNEX=0.
      SNBS=0.
      IF(FQ.LT.10.) GO TO 200
      XLMDA=299.79/FQ
      CALL INTRP(A,B,T,F,TS,FS,AA,BB,J)
      SNEX=AA*SNRT**BB
C
C     CALCULATIONS FOR SNOW BACKSCATTER.
C
      AA=0.
      BB=0.
      IA=2
      IB=2
      IF(FQ.GT.87.) IA=1
      IF(FQ.GT.82.) IB=1
      DO 100 K=1,6
      AA=AA+BSAT(K,IA)*(FQ**(K-1))
      BB=BB+BSBT(K,IB)*(FQ**(K-1))
100   CONTINUE
      AA=EXP(AA)
      BB=1.2*BB
      F1=(FQ-10.)/85.
      FCT=.367+F1*.633
```

```
      IF(FQ.GT.95.) FCT=1.                                              MMS00710
      AA=FCT*AA                                                         MMS00720
  110 IF(TK.GE.275.) J=4                                                MMS00730
      SNBS=SFCT(J-1)*AA*SNRT**BB                                        MMS00740
C                                                                       MMS00750
  200 RETURN                                                            MMS00760
C                                                                       MMS00770
      END                                                               MMS00780
```

```
      SUBROUTINE MMWFG(FD,T,FREQ,GFOG,BSFOG)
C
C*****************************************************
C
C   CALCULATES ABSORPTION DUE TO WATER  FOGS/CLOUDS,
C    AND BACKSCATTER CROSS SECTION IN M**2/M**3.
C
C   INPUTS ARE: FOG DENSITY(GM/M**3), TEMPERTURE(KELVIN),
C     FREQUENCY(GHZ).
C   OUTPUTS ARE: FOG ABSORPTION(/KM), BACKSCATTER X-SECTION(M**2 M**3).
C
C   MMFOG IS CALLED FROM MMWMOD  CALLS MMIDX SUBROUTINE.
C
C   LOCAL VARIABLES:
C      XLMDA          WAVELENGTH(MM)
C*****************************************************
C
      ICE=0
      IF(T.LT.243.) ICE=1
C COMPUTE FOG EXTINCTION
      XLMDA=10./(FREQ/29.98)
      CALL MMIDX(XLMDA,T,ICE,H20AB,H20K2)
      GFOG=18.8498*H20AB*FD/XLMDA
      BSFOG=1.162E-06*H20K2*FD**1.75/(XLMDA**4)
      RETURN
C
      END
```

```
      SUBROUTINE MMWGS(T,P,AH,FRQ,GAS)
C
C*****************************************************
C
C   SUBROUTINE COMPUTES GASEOUS ABSORPTION FROM 0 TO 1000 GHZ FOR
C     H2O VAPOR AND O2.
C
C   INPUTS INCLUDE: TEMPEATURE(KELVIN), PRESSURE(TORR), ABSOLUTE HUMIDITY
C     IN GM/M**3, FREQUENCY(GIGAHERTZ).
C   OUTPUTS ARE:  GAS ABSORPTION (1/KM)
C
C   MMWGS IS CALLED FROM NMMW, CALLS (1)MMOXY(OXYGEN ABSORPTION),
C     (2) MMH2O(H2O VAPOR ABSORPTION).
C
C   LOCAL VARIABLES:
C     DATAO2(L,J): OXYGEN LINE DATA, L=LINE NUMBER, J=TYPE:
C                  J=1 : LINE FREQUENCY(GHZ)
C                    2 : LINE STRENGTH AT 300K
C                    3 : LINE WIDTH AT 300K (GHZ/TORR)
C                    4 : INTERFERENCE PARAMETER AT 300K
C                    5 : INTERFERENCE TEMPERATURE CORRECTION
C                    6 : LINE QUANTUM PARAMETER
C     DATH2O(L,J) : H2O LINE DATA, L=LINE NUMBER, J=TYPE:
C                    1 : WAVENUMBER(1/CM)
C                    2 : STRENGTH
C                    3 : WIDTH(1/CM/TORR)
C                    4 : GROUND STATE ENERGY
C     GH2O        : H2O VAPOR ABSORPTION (1/KM)
C     GO2         : O2  ABSORPTION (1/KM)
C     PH2O        : H2O VAPOR PRESSURE(TORR)
C     WVNMB       : WAVENUMBER(1/CM)
C*****************************************************
C
      DIMENSION DATAO2(42,6), DATH2O(37,4)
      DIMENSION D1O2(42,3),D2O2(42,3),D1H2O(37,2),D2H2O(37,2)
      EQUIVALENCE (DATAO2(1,1),D1O2(1,1)),(DATAO2(1,4),D2O2(1,1))
      EQUIVALENCE (DATH2O(1,1),D1H2O(1,1)),(DATH2O(1,3),D2H2O(1,1))
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C######################################################################
      DATA D1O2/
     1 49.451,49.961,50.473,50.987,51.503,52.021,52.542,53.066,
     2 53.595,54.129,54.671,55.221,55.783,56.264,56.363,56.968,
     3 57.612,58.323,58.446,59.164,59.590,60.306,60.434,61.150,
     4 61.800,62.411,62.486,62.998,63.568,64.127,64.678,65.224,
     5 65.764,66.302,66.836,67.369,67.900,68.430,68.960,69.488,
     6 70.016,118.75,
     1 7.E-5,2.2E-4,6.E-4,1.56E-3,3.86E-3,8.99E-3,1.971E-2,.04072,
     2 .07919,.1448,.2489,.4012,.6056,.3487,.8539,1.1204,1.3595,
     3 1.515,.9251,1.5263,1.341,1.3487,1.5626,1.5899,1.4588,1.2272,
     4 .9634,.954,.6898,.4656,.2942,.1744,.0971,.0508,.025,.0116,
     5 5.08E-3,2.1E-3,8.2E-4,3.E-4,1.E-4,.5973,
     1 1.260,1.310,1.330,1.360,1.380,1.410,1.440,1.460,1.490,
     2 1.510,1.540,1.570,1.601,2.212,1.635,1.672,1.714,1.762,
     3 1.964,1.819,1.859,1.890,1.789,1.736,1.694,1.658,1.990,
     4 1.627,1.598,1.568,1.540,1.510,1.490,1.460,1.440,1.410,
     5 1.380,1.360,1.330,1.310,1.280,2.140/
      DATA D2O2/
     1 0.000,0.000,0.000,1.040,0.802,0.897,0.825,0.780,0.764,
     2 0.666,0.651,0.550,0.481,0.931,0.371,0.254,0.100,-.087,
     3 0.729,-.318,0.433,-.543,0.179,-.028,-.183,-.324,-.615,
     4 -.419,-.537,-.591,-.693,-.703,-.796,-.808,-.849,-.916,
     5 -.822,-1.05,0.000,0.000,0.000,-.054,
     1 1.00,1.00,1.00,1.38,2.04,1.69,1.91,1.88,1.90,2.01,1.95,
     2 2.11,2.13,0.89,2.36,2.66,4.20,-5.8,0.79,0.11,0.50,0.69,
     3 -.99,7.60,3.04,2.34,0.85,2.24,2.02,2.04,1.89,1.95,1.85,
     4 1.83,1.86,1.66,1.99,1.36,1.00,1.00,1.00,0.89,
     1 41.,39.,37.,35.,33.,31.,29.,27.,25.,23.,21.,19.,17.,1.,15.,13.,
     2 11.,9.,3.,7.,5.,5.,7.,9.,11.,13.,3.,15.,17.,19.,21.,23.,25.,
     3 27.,29.,31.,33.,35.,37.,39.,41.,1./
C######################################################################
```

```
      DATA D1H2O/                                                       MMG00710
     1 00.742,06.115,06.790,10.715,10.846,12.682,14.778,14.944,         MMG00720
     2 15.707,15.834,16.294,18.270,18.577,20.704,21.960,24.860,         MMG00730
     3 25.085,30.560,32.366,32.954,36.604,37.137,37.910,38.638,         MMG00740
     4 38.791,39.112,40.282,40.520,42.639,43.243,43.631,44.100,         MMG00750
     5 46.750,47.053,48.059,49.765,49.820,                              MMG00760
     1 .436E-24,.775E-22,.186E-24,.250E-23,.906E-22,.827E-21,           MMG00770
     2 .145E-22,.863E-21,.270E-22,.108E-21,.219E-22,.985E-22,           MMG00780
     3 .526E-19,.565E-21,.531E-22,.664E-22,.347E-19,.143E-20,           MMG00790
     4 .160E-20,.252E-19,.164E-18,.502E-19,.333E-21,.242E-20,           MMG00800
     5 .179E-18,.197E-21,.558E-19,.155E-21,.707E-21,.687E-21,           MMG00810
     6 .511E-22,.568E-20,.302E-21,.142E-18,.930E-21,.399E-22,           MMG00820
     7 .478E-22/                                                        MMG00830
      DATA D2H2O/                                                       MMG00840
     1 .081,.094,.095,.063,.087,.091,.050,.083,.061,.071,.075,          MMG00850
     2 .111,.107,.072,.111,.103,.102,.084,.083,.101,.097,.099,          MMG00860
     3 .094,.073,.094,.063,.093,.098,.066,.074,.060,.081,.084,          MMG00870
     4 .091,.078,.096,.097,                                             MMG00880
     1  446.512, 136.164, 134.800,1284.921, 315.780, 212.156,           MMG00890
     2 1045.069, 285.419, 742.079, 488.136, 586.482,  23.750,           MMG00900
     3   23.794, 488.110,1618.550,  69.920,  70.091, 285.219,           MMG00910
     4  383.843,  37.137, 136.761,   0.000, 172.880, 610.345,           MMG00920
     5  173.366, 888.641, 275.498,1731.890, 888.607, 842.361,           MMG00930
     6 1079.088, 508.814, 398.392, 399.459, 601.553, 100.391,           MMG00940
     7 1693.650/                                                        MMG00950
C####################################################################### MMG00960
C                                                                       MMG00970
C   COMPUTE WATER VAPOR PRESSURE, FREQUENCY IN WAVENUMBERS              MMG00980
      PH2O=AH*T*3.462977E-3                                             MMG00990
      IF(AH.LT.0.) PH2O=-PSAT(T)*AH/100.                                MMG01000
      WVNMB=FRQ/29.98                                                   MMG01010
C                                                                       MMG01020
C   COMPUTE H2O ABSORPTION                                              MMG01030
      CALL MMH2O(WVNMB,T,P,PH2O,DATH2O,GH2O)                            MMG01040
C                                                                       MMG01050
C   COMPUTE O2 ABSORPTION                                               MMG01060
      IF(FRQ.LT.140.) CALL MMOXY(FRQ,T,P,PH2O,DATAO2,GO2)               MMG01070
C                                                                       MMG01080
C   SUM ABSORPTION                                                      MMG01090
      GAS=GH2O+GO2                                                      MMG01100
C                                                                       MMG01110
C********************************************************************* MMG01120
      RETURN                                                            MMG01130
      END                                                               MMG01140
```

```
      FUNCTION PFR(T)
C       COMPUTE H2O PARTITIAN FUNCTION CORRECTIONS
C
      DIMENSION VIB(3)
      DATA VIB/3693.9,1614.5,3801.8/
C
      QJ=296./T
      QJ=QJ*SQRT(QJ)
      T1=-1.43879/296.
      T2=-1.43879/T
      T1S=1.
      T2S=1.
C
      DO 10 J=1,3
         V=VIB(J)
         T11=1.-EXP(T1*V)
         T22=1.-EXP(T2*V)
         T1S=T1S*T11
10       T2S=T2S*T22
      PFR=QJ*T2S/T1S
      RETURN
C
      END
```

```
      FUNCTION PSAT(T)
C
      DATA C1,C2,C3,C4,C5,C6,C7/-7.90298,5.02808,-1.3816E-7,11.344,
     +       8.1328E-3,-3.49149,3.005715/
      DATA D1,D2,D3,D4/-9.09718,-3.56654,.876793,.785835/
      DATA TS,TO/373.16,273.16/,CONV/.7500646/
      IF(T.LE.TO) GO TO 100
       TR=TS/T
      TRI=T/TS
      EW=C1*(TR-1.)+C2*ALOG10(TR)+C3*(10.**(C4*(1.-TRI))-1.)+
     +     C5*(10.**(C6*(TR-1.))-1.)+C7
      GO TO 200
100   TR=TO/T
      TRI=T/TO
      EW=D1*(TR-1.)+D2*ALOG10(TR)+D3*(1.-TRI)+D4
200   PSAT=(10.**EW)*CONV
      RETURN
      END
```

```
      FUNCTION SUPK(A,B,C)
C        COMPUTES THE SUPER KINETIC LINE PROFILE FACTOR.
C
      PI=3.14159265
      XNORM=.998776
      VM=10.*C
      X=ABS((B*B-A*A)/(2.*C*A))
      IF(ABS(B-A).GT.VM) GO TO 10
C
      SUPK=XNORM/(PI*C)/(X*X+1.)
      RETURN
C
10    CXI=10.*(B+10.*C/2.)/(B+10.*C)
      CXI=((CXI**1.88)+1.)/((CXI*CXI)+1.)
      SUPK=XNORM*CXI/(PI*C)/((X**1.88)+1.)
      RETURN
      END
```

```
      SUBROUTINE CLTRAN(CTRANS,WAVE,IRUN,IERR)
      DIMENSION TAU(20)
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,
     +       WNDVEL,WNDDIR,IPASCT
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWY,ILCTYP(10),ICCTYP(10)
      COMMON /CYL/XC,YC,RADIUS
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)
     +,RADICL(10)
      COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)
C-----------------------------------------------------------------------
C** DATA IS READ FROM INPUT RECORDS AND THEN TRANSFERRED TO CLTRAN
C** BY SUBROUTINE CLREAD. INPUT TO CLTRAN IS CARD ORDER-INDEPENDENT,
C** WITH A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH INPUT RECORD.
C** THE ONLY EXCEPTION TO THIS ORDER-INDEPENDENCE IS THE GO SENTINEL
C** CARD, WHICH MUST BE THE LAST RECORD READ. ALL CARDS ARE READ IN
C** UNDER THE FORMAT (A4,1X,5(E10.5,1X)). THE IDENTIFICATION AND MEANING
C** OF EACH INPUT RECORD ARE AS FOLLOWS :
C-----------------------------------------------------------------------
C      CARD IDENTIFIER : SEEK
C      VARIABLES READ : XS,YS,ZS
C        XS,YS,ZS = POSITION COORDINATES OF SEEKER (OR RECEIVER) (KM)
C-----------------------------------------------------------------------
C      CARD IDENTIFIER : TARG
C      VARIABLES READ : XT,YT,ZT
C        XT,YT,ZT = POSITION COORDINATES OF TARGET  (KM)
C-----------------------------------------------------------------------
C
C** THE FOLLOWING FOUR CARDS REPRESENT THE STRATIFORM CLOUDS TREATED
C** BY CLTRAN. THEY ARE TREATED AS INFINITE LAYERS IN A PLANE-PARALLEL
C** ATMOSPHERE.
C
C-----------------------------------------------------------------------
C      CARD IDENTIFIER : CLST  (STRATUS CLOUD TYPE)
C      VARIABLES READ : ZLBASE,ZLTHIC
C        ZLBASE = HEIGHT OF CLOUD BASE  (KM)
C        ZLTHIC = VERTICAL THICKNESS OF CLOUD LAYER  (KM)
C-----------------------------------------------------------------------
C      CARD IDENTIFIER : CLAS  (ALTOSTRATUS CLOUD TYPE)
C      VARIABLES READ : ZLBASE,ZLTHIC
C-----------------------------------------------------------------------
C      CARD IDENTIFIER : CLNS  (NIMBOSTRATUS CLOUD TYPE)
C      VARIABLES READ : ZLBASE,ZLTHIC
C-----------------------------------------------------------------------
C      CARD IDENTIFIER : CLSC  (STRATOCUMULUS CLOUD TYPE)
C      VARIABLES READ : ZLBASE,ZLTHIC
C-----------------------------------------------------------------------
C
C** THE NEXT TWO CARDS REPRESENT THE CUMULUS CLOUD TYPES ADDRESSED
C** BY CLTRAN. CLOUDS OF THIS KIND ARE MODELLED AS CYLINDERS WHICH
C** HAVE VERTICAL SYMMETRY AXES. THESE TYPES ARE REPRESENTED BY
C** THE FOLLOWING RECORDS :
C
C-----------------------------------------------------------------------
C      CARD IDENTIFIER : CLCH  (CUMULUS HUMILIS CLOUD TYPE)
C      VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD
C        ZCBASE = HEIGHT OF CLOUD CYLINDER'S LOWER BASE (KM)
C        ZCTHIC = VERTICAL THICKNESS OF CLOUD CYLINDER  (KM)
C        RADICL = RADIUS OF CLOUD CYLINDER  (KM)
C        XCLOUD = X-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM)
C        YCLOUD = Y-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM)
C-----------------------------------------------------------------------
C      CARD IDENTIFIER : CLCC  (CUMULUS CONGESTUS CLOUD TYPE)
C      VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD
C-----------------------------------------------------------------------
C
C** THE FOLLOWING CARD MUST BE THE LAST RECORD READ :
```

```
C                                                                       CLT00690
C-----------------------------------------------------------------------CLT00700
C     CARD IDENTIFIER : GO                                              CLT00710
C     VARIABLES READ  : NONE                                            CLT00720
C-----------------------------------------------------------------------CLT00730
      IF(IRUN.GT.1) GO TO 5                                             CLT00740
      NLAYER=0                                                          CLT00750
      NCCLDS=0                                                          CLT00760
    5 CONTINUE                                                          CLT00770
      IWV=0                                                             CLT00780
      ZBMIN=999.
      FOGPRB=0.
      IF(WAVE.GT.0.20.AND.WAVE.LT.2.00) IWV=1                           CLT00790
      IF(WAVE.GT.3.00.AND.WAVE.LT.5.00) IWV=2                           CLT00800
      IF(WAVE.GT. 8.0.AND.WAVE.LT.12.0) IWV=3                           CLT00810
      IF(IWV.NE.0) GO TO 50                                             CLT00820
      IERR=1                                                            CLT00830
      WRITE(IOOUT,22)                                                   CLT00840
   22 FORMAT(1H0,20X,94H***CLTRAN ERROR*** INPUT WAVELENGTH DOES NOT LIECLT00850
     + WITHIN ALLOWABLE LIMITS, EXECUTION TERMINATED /)                 CLT00860
      GO TO 900                                                         CLT00870
   50 CALL CLREAD(NLAYER,NCCLDS,IERR)                                   CLT00880
      IF(IERR.EQ.1) GO TO 900                                           CLT00890
      ISLTUP=0                                                          CLT00900
      ISLTDN=0                                                          CLT00910
      IHORIZ=0                                                          CLT00920
      IVERT=0                                                           CLT00930
C                                                                       CLT00940
C**** DETERMINE SENSE OF L-O-S SLOPE FROM SEEKER'S POINT OF VIEW        CLT00950
C                                                                       CLT00960
      TESVER=(XS-XT)**2+(YS-YT)**2                                      CLT00970
      IF(TESVER.EQ.0.0)IVERT=1                                          CLT00980
      IF(IVERT.EQ.1)GO TO 300                                           CLT00990
      IF(ZS-ZT)200,210,220                                              CLT01000
  200 ISLTUP=1                                                          CLT01010
      GO TO 300                                                         CLT01020
  210 IHORIZ=1                                                          CLT01030
      GO TO 300                                                         CLT01040
  220 ISLTDN=1                                                          CLT01050
  300 CONTINUE                                                          CLT01060
      IF(IHORIZ.EQ.1)GO TO 310                                          CLT01070
C                                                                       CLT01080
C**** COMPUTE L-O-S SLOPES IN X-Z AND Y-Z VERTICAL PLANES               CLT01090
C                                                                       CLT01100
      XIX=(XS-XT)/(ZS-ZT)                                               CLT01110
      XIY=(YS-YT)/(ZS-ZT)                                               CLT01120
  310 CONTINUE                                                          CLT01130
C                                                                       CLT01140
C**** STRATIFORM CLOUD BLOCK                                            CLT01150
C                                                                       CLT01160
      IF(NLAYER.EQ.0)GO TO 500                                          CLT01170
C                                                                       CLT01180
C**** UTILIZE DEFAULT BASE OR THICKNESS VALUES IF NECESSARY             CLT01190
C                                                                       CLT01200
      CALL DEFSET(1,NLAYER)                                             CLT01210
      DO 400 N=1,NLAYER                                                 CLT01220
      NLINT(N)=0                                                        CLT01230
      Z0(N)=ZLBASE(N)                                                   CLT01240
      Z1(N)=Z0(N)+ZLTHIC(N)                                             CLT01250
      ZBAS=Z0(N)                                                        CLT01260
      ZTOP=Z1(N)                                                        CLT01270
      ICL=ILCTYP(N)                                                     CLT01280
      IF(ZBMIN.LT.ZLBASE(N)) GO TO 320
      ZBMIN=ZLBASE(N)
      FOGPRB=FLOAT(ICL)
  320 CONTINUE
C                                                                       CLT01290
C**** DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CLOUD LAYER (IF        CLT01300
C**** THERE ARE ANY): (X1(N),Y1(N),Z1(N))= UPPER INTERSECTION POINT,    CLT01310
C**** (X0(N),Y0(N),Z0(N))= LOWER INTERSECTION POINT                     CLT01320
```

```
C
      CALL LAYRXY(XIX,XIY,N)
      IF(TESALL.LE.0.0)GO TO 350
      NLINT(N)=N
C
C**** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH
C
      CALL CLEXTN(TAUN,N)
      TAU(N)=TAUN
      ATTL(N)=SQRT((X1(N)-X0(N))**2+(Y1(N)-Y0(N))**2+(Z1(N)-Z0(N))**2)
      GO TO 400
C
C**** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH
C**** IS NOW SET TO ZERO
C
  350 TAU(N)=0.0
      ATTL(N)=0.0
  400 CONTINUE
C
C**** END STRATIFORM CLOUD BLOCK
C
C**** CUMULIFORM CLOUD BLOCK
C
  500 IF(NCCLDS.EQ.0)GO TO 805
C
C**** UTILIZE DEFAULT BASE, THICKNESS, OR RADIUS IF NECESSARY
C
      CALL DEFSET(2,NCCLDS)
      DO 800 N=1,NCCLDS
      NN=NLAYER+N
      NCINT(N)=0
      Z0(NN)=ZCBASE(N)
      Z1(NN)=Z0(NN)+ZCTHIC(N)
      ICL=ICCTYP(N)
      IF(ZBMIN.LT.ZCBASE(N)) GO TO 520
      ZBMIN=ZCBASE(N)
      FOGPRB=FLOAT(ICL)
  520 CONTINUE
      RADIUS=RADICL(N)
      XC=XCLOUD(N)
      YC=YCLOUD(N)
      ZBAS=Z0(NN)
      ZTOP=Z1(NN)
      TESARG=0.0
      TESVT=0.0
C
C**** DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CUMULUS CLOUD CYLINDER
C**** (IF THERE ARE ANY): (X1(N),Y1(N),Z1(N))= UPPER INTERSECTION POINT,
C**** (X0(N),Y0(N),Z0(N))= LOWER INTERSECTION POINT
C
      CALL CYLXY(XIX,XIY,NN)
      IF(TESALL.LE.0.0.OR.TESARG.LT.0.0.OR.TESVT.LT.0.0)GO TO 650
      NCINT(N)=N
C
C**** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH
C
      CALL CLEXTN(TAUN,NN)
      TAU(NN)=TAUN
      ATTL(NN)=SQRT((X1(NN)-X0(NN))**2+(Y1(NN)-Y0(NN))**2+
     +(Z1(NN)-Z0(NN))**2)
      GO TO 800
C
C**** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH
C**** IS NOW SET TO ZERO
C
  650 TAU(NN)=0.0
      ATTL(NN)=0.0
  800 CONTINUE
C
C**** END CUMULIFORM CLOUD BLOCK
```

```
C
  805 CONTINUE
C**** DETERMINE CUMULATIVE CLOUD OPTICAL DEPTH AND TRANSMITTANCE
      TAUTOT=0.0
      NN=NLAYER+NCCLDS
      IF(NN.EQ.0)GO TO 820
      DO 810 N=1,NN
  810 TAUTOT=TAUTOT+TAU(N)
  820 CTRANS=EXP(-TAUTOT)
  850 CALL LISOUT(NN,CTRANS,TAUTOT,TAU,NLAYER,IRUN)
  900 RETURN
      END
```

```
      SUBROUTINE LAYRXY(XIX,XIY,IN)
      LOGICAL INSEEK,INTARG
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)
C**** CHECK LOS TO SEE IF IT INTERSECTS THE CLOUD LAYER
      UP1=ISLTUP*(ZS-Z1(IN))
      UP2=ISLTUP*(Z0(IN)-ZT)
      DN1=ISLTDN*(ZT-Z1(IN))
      DN2=ISLTDN*(Z0(IN)-ZS)
      HOR12=IHORIZ*(ZS-Z1(IN))*(ZS-Z0(IN))*(-1.0)
      UPVER1=IVERT*(ZT-ZS)*(ZS-Z1(IN))
      UPVER2=IVERT*(ZT-ZS)*(Z0(IN)-ZT)
      TESALL=UP1*UP2+DN1*DN2+HOR12+UPVER1*UPVER2
      IF(TESALL.LT.0.0)RETURN
      IF(IHORIZ.EQ.1)RETURN
C
C**** COMPUTE X,Y INTERSECTIONS OF CLOUD PLANES AND LOS
C**** ALSO, CHECK FOR THE CASE WHERE EITHER THE SEEKER
C**** OR TARGET IS INSIDE OF THE CLOUD
C
      INSEEK=ZS.LE.Z1(IN).AND.ZS.GE.Z0(IN)
      INTARG=ZT.LE.Z1(IN).AND.ZT.GE.Z0(IN)
      IF(ISLTUP.EQ.1.AND.INTARG)Z1(IN)=ZT
      IF(ISLTDN.EQ.1.AND.INTARG)Z0(IN)=ZT
      IF(ISLTUP.EQ.1.AND.INSEEK)Z0(IN)=ZS
      IF(ISLTDN.EQ.1.AND.INSEEK)Z1(IN)=ZS
      X1(IN)=XS+XIX*(Z1(IN)-ZS)
      Y1(IN)=YS+XIY*(Z1(IN)-ZS)
      X0(IN)=XS+XIX*(Z0(IN)-ZS)
      Y0(IN)=YS+XIY*(Z0(IN)-ZS)
      RETURN
      END
```

```
      SUBROUTINE CYLXY(XIX,XIY,IN)
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
      COMMON /CYL/XC,YC,RADIUS
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)
C**** FIRST DETERMINE INTERSECTION PTS OF LOS WITH UPPER
C**** AND LOWER BASE PLANES OF CLOUD CYLINDER
      TESALL=0.0
      CALL LAYRXY(XIX,XIY,IN)
      IF(TESALL.LT.0.0)RETURN
      X0LAYR=X0(IN)
      Y0LAYR=Y0(IN)
      Z0LAYR=Z0(IN)
      X1LAYR=X1(IN)
      Y1LAYR=Y1(IN)
      Z1LAYR=Z1(IN)
C**** CALCULATE NEXT THE INTERSECTION PTS OF THE LOS
C**** WITH THE SURFACE OF AN INFINITE VERTICAL CYLINDER WITH
C**** THE SAME RADIUS (RADIUS)AND LATERAL POSITION (XC,YC) AS THE CLOUD
      IF(IVERT.NE.1)GO TO 40
      VERTES=SQRT((XT-XC)**2+(YT-YC)**2)
      TESVT=RADIUS-VERTES
      IF(VERTES.LT.RADIUS)GO TO 400
      RETURN
   40 IF((XT-XS).EQ.0.0)GO TO 60
      ALPHA=(YT-YS)/(XT-XS)
      A2=ALPHA**2
      C2=1.0+A2
      C1=2.0*(ALPHA*(YT-YC-ALPHA*XT)-XC)
      C0=XC**2+(YT-YC-ALPHA*XT)**2-RADIUS**2
      TESARG=C1**2-4.0*C2*C0
      IF(TESARG.LT.0.0)RETURN
      XP=(-C1+SQRT(TESARG))/(2.0*C2)
      XM=(-C1-SQRT(TESARG))/(2.0*C2)
      YP=YT+ALPHA*(XP-XT)
      YM=YT+ALPHA*(XM-XT)
      IF(IHORIZ.EQ.1)GO TO 300
      ZM=ZS+(1.0/XIX)*(XM-XS)
      ZP=ZS+(1.0/XIX)*(XP-XS)
C**** CHECK FOR SKEW MISS OF CLOUD
      TOPP=ZP-ZTOP
      TOPM=ZM-ZTOP
      BASP=ZBAS-ZP
      BASM=ZBAS-ZM
      IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50
      TESALL=-1.0
      RETURN
   60 XP=XT
      TESARG=RADIUS**2-(XP-XT)**2
      IF(TESARG.LT.0.0)RETURN
      YP=YC+SQRT(TESARG)
      XM=XT
      YM=YC+SQRT(TESARG)
      IF(IHORIZ.EQ.1)GO TO 300
      ZM=ZS+(1.0/XIY)*(YM-YS)
      ZP=ZS+(1.0/XIY)*(YP-YS)
C**** CHECK FOR SKEW MISS OF CLOUD
      TOPP=ZP-ZTOP
      TOPM=ZM-ZTOP
      BASP=ZBAS-ZP
      BASM=ZBAS-ZM
      IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50
      TESALL=-1.0
      RETURN
   50 IF(ZP.LT.ZM)GO TO 100
      X0CYL=XM
      Y0CYL=YM
      Z0CYL=ZM
      X1CYL=XP
```

```
      Y1CYL=YP
      Z1CYL=ZP
      GO TO 200
  100 X0CYL=XP
      Y0CYL=YP
      Z0CYL=ZP
      X1CYL=XM
      Y1CYL=YM
      Z1CYL=ZM
  200 CONTINUE
      GO TO 450
  300 X0(IN)=XM
      Y0(IN)=YM
      Z0(IN)=ZT
      X1(IN)=XP
      Y1(IN)=YP
      Z1(IN)=ZT
      GO TO 500
  400 X0(IN)=XT
      Y0(IN)=YT
      X1(IN)=XT
      Y1(IN)=YT
      GO TO 500
  450 IF(Z0CYL.LT.Z0LAYR)GO TO 460
      X0(IN)=X0CYL
      Y0(IN)=Y0CYL
      Z0(IN)=Z0CYL
      GO TO 470
  460 X0(IN)=X0LAYR
      Y0(IN)=Y0LAYR
      Z0(IN)=Z0LAYR
  470 IF(Z1CYL.GT.Z1LAYR)GO TO 480
      X1(IN)=X1CYL
      Y1(IN)=Y1CYL
      Z1(IN)=Z1CYL
      GO TO 500
  480 X1(IN)=X1LAYR
      Y1(IN)=Y1LAYR
      Z1(IN)=Z1LAYR
  500 RETURN
      END
```

```
      SUBROUTINE CLEXTN(TAUN,IN)
      DIMENSION AA(42),BB(42),CC(42),A(7,6,3)
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
      EQUIVALENCE (AA(1),A(1,1,1)),(BB(1),A(1,1,2)),(CC(1),A(1,1,3))
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV
      DATA AA/5.3744E-2,2.9313E-4,-1.1300E-6,1.6228E-9,-1.0421E-12,
     +2.4192E-16,0.0000E-00,8.4451E-2,1.6549E-4,-4.7300E-7,
     +4.4334E-10,-1.3851E-13,-2.0644E-17,1.2569E-20,1.0240E-1,
     +4.8678E-5,-2.2195E-7,1.4506E-10,5.9393E-14,-8.7250E-17,
     +2.1383E-20,3.6775E-2,1.6160E-4,-6.4666E-7,9.5298E-10,
     +-6.4023E-13,1.6916E-16,-7.9924E-21,1.8410E-2,6.6870E-4,
     +-2.8406E-6,5.4478E-9,-5.4854E-12,2.7932E-15,-5.8271E-19,
     +3.3458E-2,1.3098E-4,-4.1528E-7,6.1166E-10,-4.6681E-13,
     +1.7383E-16,-2.5043E-20/
      DATA BB/7.5099E-2,3.2061E-4,-1.7060E-6,3.3538E-9,-3.2719E-12,
     +1.5721E-15,-3.0123E-19,1.1808E-1,2.2387E-4,-8.7996E-7,
     +1.1298E-9,-6.6823E-13,1.7514E-16,-1.5665E-20,1.4155E-1,
     +-5.0592E-5,-1.2280E-7,1.2715E-10,2.7470E-14,-6.6940E-17,
     +1.7734E-20,4.9533E-2,2.0904E-4,-1.1626E-6,2.3531E-9,
     +-2.3463E-12,1.1472E-15,-2.2267E-19,3.8315E-3,1.1837E-3,
     +-5.1096E-6,9.4660E-9,-8.6426E-12,3.6778E-15,-5.5775E-19,
     +4.1534E-2,2.0220E-4,-8.1463E-7,1.3590E-9,-1.1142E-12,
     +4.3867E-16,-6.6426E-20/
      DATA CC/1.1269E-2,2.8659E-4,-6.0210E-8,-1.5274E-9,2.7747E-12,
     +-1.8946E-15,4.5539E-19,1.9856E-2,1.2292E-4,-8.4479E-8,
     +8.6563E-12,0.0000E-0,0.0000E-0,0.0000E-0,3.1907E-2,
     +1.9632E-4,-1.1114E-7,-3.3987E-10,5.2528E-13,-2.7799E-16,
     +5.0469E-20,6.8522E-3,1.5362E-4,7.5813E-8,-1.1430E-9,
     +1.8885E-12,-1.2477E-15,2.9522E-19,8.5792E-4,6.4122E-5,
     +7.7271E-7,-2.9750E-9,4.4014E-12,-3.0626E-15,8.1541E-19,
     +3.7151E-3,1.4919E-4,-9.3486E-8,-1.2183E-10,1.8733E-13,
     +-9.2167E-17,1.5939E-20/
      TAUN=0.0
      ZZA=Z0(IN)-ZBAS
      ZZB=Z1(IN)-ZBAS
      VERDIS=ZZB-ZZA
      HORDIS=SQRT((X1(IN)-X0(IN))**2+(Y1(IN)-Y0(IN))**2)
      EL=SQRT(HORDIS**2+VERDIS**2)
      IF(EL.EQ.0.0)RETURN
      IF(IHORIZ.EQ.1)GO TO 200
      XI=VERDIS/EL
      ELA=ZZA*1000.0
      ELB=(XI*EL+ZZA)*1000.0
      POLYA=0.0
      POLYB=0.0
      DO 100 N=1,7
      EN=FLOAT(N)
      AN=A(N,ICL,IWV)
      TERMB=(1.0/(XI*EN))*AN*ELB**N
      TERMA=(1.0/(XI*EN))*AN*ELA**N
      POLYA=POLYA+TERMA
  100 POLYB=POLYB+TERMB
      TAUN=POLYB-POLYA
      GO TO 300
  200 ZZA=ZZA*1000.0
      DO 250 N=1,7
      TERMK=A(N,ICL,IWV)*HORDIS*1000.0*ZZA**(N-1)
  250 TAUN=TAUN+TERMK
  300 RETURN
      END
```

```
      SUBROUTINE LISOUT(NN,CTRANS,TAUTOT,TAU,NLAYER,IRUN)
C
C**** OUTPUT CONTROL ROUTINE
C
      DIMENSION ALPH(6),T(20),TAU(20)
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
      COMMON /PATHL/ X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)
      COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)
     +,RADICL(10)
      COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DATA ALPH/2HST,2HAS,2HNS,2HSC,2HCH,2HCC/
      IF(IWV.EQ.1)WVL=0.55
      IF(IWV.EQ.2)WVL=3.80
      IF(IWV.EQ.3)WVL=10.60
      IF(IRUN.LT.2) WRITE(IOOUT,150)WVL
      IF(IRUN.GT.1) WRITE(IOOUT,160)WVL
  150 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS /)
  160 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS /)
      IF(IHORIZ.EQ.1)WRITE(IOOUT,200)
      IF(IVERT.EQ.1)WRITE(IOOUT,210)
      IF(ISLTUP.EQ.1)WRITE(IOOUT,220)
      IF(ISLTDN.EQ.1)WRITE(IOOUT,230)
  200 FORMAT(1H0,40X,28HLINE-OF-SIGHT IS HORIZONTAL //)
  210 FORMAT(1H0,40X,26HLINE-OF-SIGHT IS VERTICAL //)
  220 FORMAT(1H0,40X,28HLINE-OF-SIGHT SLANTS UPWARD //)
  230 FORMAT(1H0,40X,30HLINE-OF-SIGHT SLANTS DOWNWARD //)
      PTHLEN=SQRT((XS-XT)**2+(YS-YT)**2+(ZS-ZT)**2)
      WRITE(IOOUT,300)PTHLEN
  300 FORMAT(1H0,40X,29HTOTAL LINE-OF-SIGHT LENGTH = ,F7.3,4H KM )
      ATTLEN=0.0
      IF(TAUTOT.EQ.0.0)GO TO 780
      DO 310 N=1,NN
  310 ATTLEN=ATTLEN+ATTL(N)
      WRITE(IOOUT,400)ATTLEN
  400 FORMAT(1H0,40X,50HTOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD
     += ,F7.3,4H KM )
      WRITE(IOOUT,500)TAUTOT
  500 FORMAT(1H0,40X,22HTOTAL OPTICAL DEPTH = ,F7.2/)
      WRITE(IOOUT,600)CTRANS
  600 FORMAT(1H0,40X,36HTRANSMITTANCE ALONG LINE-OF-SIGHT = ,E11.5,//)
      WRITE(IOOUT,605)
  605 FORMAT(1H0,40X,50HSEEKER COORDINATES (KM)   TARGET COORDINATES (KM
     +) )
      WRITE(IOOUT,606)
  606 FORMAT(1H ,40X,23HXSEEKER YSEEKER ZSEEKER,3X,23HXTARGET YTARGET ZT
     +ARGET)
      WRITE(IOOUT,607)
  607 FORMAT(1H ,40X,3(8H-------  ),2X,3(8H-------  ))
      WRITE(IOOUT,608)XS,YS,ZS,XT,YT,ZT
  608 FORMAT(1H0,40X,3(F6.3,2X),2X,3(F6.3,2X))
      WRITE(IOOUT,613)
  613 FORMAT(1H ,//)
      WRITE(IOOUT,615)
  615 FORMAT(1H0,40X,62HCLOUD TYPE    LINE-OF-SIGHT  INTERSECTION  COORD
     +INATES   (KM) )
      WRITE(IOOUT,620)
  620 FORMAT(1H ,40X,10H/ID NUMBER,4X,47HXUPPER  YUPPER  ZUPPER   XLOWER
     +  YLOWER  ZLOWER)
      WRITE(IOOUT,625)
  625 FORMAT(1H ,40X,10H----------,4X,3(8H------  ),1X,3(8H------  ))
      IF(NLAYER.EQ.0)GO TO 685
      DO 670 N=1,NLAYER
      IND=ILCTYP(N)
      IF(NLINT(N).EQ.0)GO TO 670
      WRITE(IOOUT,680)ALPH(IND),N,X1(N),Y1(N),Z1(N),X0(N),Y0(N),Z0(N)
  670 CONTINUE
  680 FORMAT(1H0,42X,A2,1H/,I2,6X,3(F7.3,1X),1X,3(F7.3,1X))
```

```
  685 NLO=NLAYER+1
      IF(NLAYER.EQ.NN)GO TO 699
      LL=0
      DO 690 N=NLO,NN
      LL=LL+1
      IND=ICCTYP(LL)
      IF(NCINT(LL).EQ.0)GO TO 690
      WRITE(IOOUT,680)ALPH(IND),N,X1(N),Y1(N),Z1(N),X0(N),Y0(N),Z0(N)
  690 CONTINUE
      WRITE(IOOUT,613)
      WRITE(IOOUT,691)
  691 FORMAT(1H0,40X,59HCLOUD TYPE   CUMULUS CENTER POSITIONS (FOR CC OR
     + CH TYPES) )
      WRITE(IOOUT,692)
  692 FORMAT(1H ,40X,10H/ID NUMBER,6X,26HXCLOUD (KM)    YCLOUD (KM))
      WRITE(IOOUT,693)
  693 FORMAT(1H ,40X,10H----------,6X,26H-----------   ------------)
      LL=0
      DO 695 N=NLO,NN
      LL=LL+1
      IND=ICCTYP(LL)
  695 WRITE(IOOUT,696)ALPH(IND),N,XCLOUD(LL),YCLOUD(LL)
  696 FORMAT(1H0,42X,A2,1H/,I2,10X,F7.3,8X,F7.3)
  699 CONTINUE
      WRITE(IOOUT,613)
      DO 610 N=1,NN
  610 T(N)=EXP(-TAU(N))
      IF(NCCLDS.GT.0) WRITE(IOOUT,700)
      IF(NCCLDS.LT.1) WRITE(IOOUT,705)
  700 FORMAT(1H1,20X,11HCLOUD TYPE ,2X,15HHEIGHT OF BASE ,2X,
     +10HTHICKNESS ,2X,16HRADIUS OF CLOUD ,2X,14HOPTICAL DEPTH ,2X,
     +14HTRANSMITTANCE )
  705 FORMAT(1H0,20X,11HCLOUD TYPE ,2X,15HHEIGHT OF BASE ,2X,
     +10HTHICKNESS ,2X,16HRADIUS OF CLOUD ,2X,14HOPTICAL DEPTH ,2X,
     +14HTRANSMITTANCE )
      WRITE(IOOUT,710)
  710 FORMAT(1H ,20X,10H/ID NUMBER,9X,4H(KM),9X,4H(KM),12X,4H(KM),9X,
     +11HALONG L-O-S,5X,11HALONG L-O-S)
      WRITE(IOOUT,720)
  720 FORMAT(1H ,20X,10H----------,3X,14H--------------,3X,9H---------,
     +3X,15H---------------,3X,13H-------------,3X,13H-------------)
      IF(NLAYER.EQ.0)GO TO 735
      BLANK=0.0
      DO 730 N=1,NLAYER
      IND=ILCTYP(N)
  730 WRITE(IOOUT,760)ALPH(IND),N,ZLBASE(N),ZLTHIC(N),BLANK,TAU(N),T(N)
  735 NLO=NLAYER+1
  740 IF(NN.EQ.NLAYER)GO TO 800
      LL=0
      DO 750 N=NLO,NN
      LL=LL+1
      IND=ICCTYP(LL)
  750 WRITE(IOOUT,760)ALPH(IND),N,ZCBASE(LL),ZCTHIC(LL),RADICL(LL),
     +TAU(N),T(N)
  760 FORMAT(1H0,22X,A2,1H/,I2,10X,F7.3,7X,F7.3,8X,F7.3,10X,F7.2,7X,
     +E11.5)
      GO TO 800
  780 WRITE(IOOUT,790)
  790 FORMAT(1H0,20X,59HNO CLOUD OBSCURATION : L-O-S DOES NOT INTERSECT
     +ANY CLOUDS //)
  800 RETURN
      END
```

```
      SUBROUTINE DEFSET(ISTEP,NMAX)                                    DEF00010
C                                                                      DEF00020
C**** THIS ROUTINE RESETS THE CLOUD BASE HEIGHT, THICKNESS,             DEF00030
C**** AND RADIUS VALUES TO THE NEAREST REALISTIC BOUNDARIES            DEF00040
C**** IF THEY DO NOT LIE CLOSE TO THE RANGES SPECIFIED IN              DEF00050
C**** R. D. H. LOW'S PAPER.                                            DEF00060
C                                                                      DEF00070
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)           DEF00080
      COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)        DEF00090
     +,RADICL(10)                                                      DEF00100
      IF(ISTEP.NE.1)GO TO 10                                           DEF00110
      DO 5 N=1,NMAX                                                    DEF00120
      ITYPE=ILCTYP(N)                                                  DEF00130
      IF(ITYPE.EQ.2)GO TO 3                                            DEF00140
    1 IF(ZLBASE(N).LT.0.1)ZLBASE(N)=0.1                                DEF00150
      IF(ZLBASE(N).GT.1.5)ZLBASE(N)=1.5                                DEF00160
      IF(ITYPE.EQ.3)GO TO 4                                            DEF00170
    2 IF(ZLTHIC(N).LT.0.2)ZLTHIC(N)=0.2                                DEF00180
      IF(ZLTHIC(N).GT.1.0)ZLTHIC(N)=1.0                                DEF00190
      GO TO 5                                                          DEF00200
    3 IF(ZLBASE(N).LT.2.0)ZLBASE(N)=2.0                                DEF00210
      IF(ZLBASE(N).GT.5.0)ZLBASE(N)=5.0                                DEF00220
    4 IF(ZLTHIC(N).LT.1.0)ZLTHIC(N)=1.0                                DEF00230
      IF(ZLTHIC(N).GT.4.0)ZLTHIC(N)=4.0                                DEF00240
    5 CONTINUE                                                         DEF00250
   10 IF(ISTEP.NE.2)GO TO 20                                           DEF00260
      DO 15 N=1,NMAX                                                   DEF00270
      IF(ZCBASE(N).LT.0.8)ZCBASE(N)=0.8                                DEF00280
      IF(ZCBASE(N).GT.1.5)ZCBASE(N)=1.5                                DEF00290
      IF(ZCTHIC(N).LT.0.2)ZCTHIC(N)=0.2                                DEF00300
      IF(ZCTHIC(N).GT.5.0)ZCTHIC(N)=5.0                                DEF00310
      IF(RADICL(N).LT.0.05)RADICL(N)=0.05                              DEF00320
      IF(RADICL(N).GT.0.6)RADICL(N)=0.6                                DEF00330
   15 CONTINUE                                                         DEF00340
   20 RETURN                                                           DEF00350
      END                                                              DEF00360
```

```
      SUBROUTINE CLREAD(NLAYER,NCCLDS,IERR)                               CLR00010
      DIMENSION IALPHA(18),DATELT(5)                                      CLR00020
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)                 CLR00030
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)          CLR00040
      COMMON /GEOMET/PTS(15),IGEOSW                                       CLR00050
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)              CLR00060
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCLR00070
      COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)           CLR00080
     +,RADICL(10)                                                         CLR00090
      COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)             CLR00100
      DATA IALPHA/2HCL,2HST,2HCL,2HAS,2HCL,2HNS,2HCL,2HSC,2HCL,2HCH,      CLR00110
     +2HCL,2HCC,2HSE,2HEK,2HTA,2HRG,2HGO,2H  /                            CLR00120
      ISFLAG=0                                                            CLR00130
      ITFLAG=0                                                            CLR00140
      IF(IGEOSW.NE.1)GO TO 60                                             CLR00150
      XS=PTS(4)                                                           CLR00160
      YS=PTS(5)                                                           CLR00170
      ZS=PTS(6)                                                           CLR00180
      XT=PTS(1)                                                           CLR00190
      YT=PTS(2)                                                           CLR00200
      ZT=PTS(3)                                                           CLR00210
   60 CONTINUE                                                            CLR00220
      DO 900 N=1,23                                                       CLR00230
      READ(IOIN,100)IALPH,IALP,(DATELT(I),I=1,5)                          CLR00240
  100 FORMAT(2A2,1X,5(E10.5,1X))                                          CLR00250
      INDEX=10                                                            CLR00260
      DO 200 K=1,17,2                                                     CLR00270
      IF((IALPH.EQ.IALPHA(K)).AND.(IALP.EQ.IALPHA(K+1))) GO TO 50         CLR00280
      GO TO 200
   50 REALK=FLOAT(K)
      RQT=REALK/2.0
      KJ=IFIX(RQT)+1
      INDEX=KJ
      GO TO 300                                                           CLR00300
  200 CONTINUE                                                            CLR00310
  300 IF(INDEX.LT.7)GO TO (350,350,350,350,400,400),INDEX                 CLR00320
      INM6=INDEX-6                                                        CLR00330
      GO TO (500,600,999,997),INM6                                        CLR00340
  350 NLAYER=NLAYER+1                                                     CLR00350
      IF(NLAYER.GT.10) GO TO 993                                          CLR00360
      ILCTYP(NLAYER)=INDEX                                                CLR00370
      ZLBASE(NLAYER)=DATELT(1)                                            CLR00380
      ZLTHIC(NLAYER)=DATELT(2)                                            CLR00390
      GO TO 900                                                           CLR00400
  400 NCCLDS=NCCLDS+1                                                     CLR00410
      IF(NCCLDS.GT.10) GO TO 995                                          CLR00420
      ICCTYP(NCCLDS)=INDEX                                                CLR00430
      ZCBASE(NCCLDS)=DATELT(1)                                            CLR00440
      ZCTHIC(NCCLDS)=DATELT(2)                                            CLR00450
      RADICL(NCCLDS)=DATELT(3)                                            CLR00460
      XCLOUD(NCCLDS)=DATELT(4)                                            CLR00470
      YCLOUD(NCCLDS)=DATELT(5)                                            CLR00480
      GO TO 900                                                           CLR00490
  500 IF(ISFLAG.EQ.1) GO TO 997                                           CLR00500
      XS=DATELT(1)                                                        CLR00510
      YS=DATELT(2)                                                        CLR00520
      ZS=DATELT(3)                                                        CLR00530
      ISFLAG=1                                                            CLR00540
      GO TO 900                                                           CLR00550
  600 IF(ITFLAG.GT.1) GO TO 997                                           CLR00560
      XT=DATELT(1)                                                        CLR00570
      YT=DATELT(2)                                                        CLR00580
      ZT=DATELT(3)                                                        CLR00590
      ITFLAG=1                                                            CLR00600
  900 CONTINUE                                                            CLR00610
      GO TO 999                                                           CLR00620
  993 IERR=1                                                              CLR00630
      WRITE(IOOUT,994)                                                    CLR00640
  994 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF STRATIFORM CLOUDS ICLR00650
     +NPUT EXCEEDS THE LIMIT OF 10 //)                                    CLR00660
```

```
      GO TO 999                                                         CLR00670
  995 IERR=1                                                            CLR00680
      WRITE(IOOUT,996)                                                  CLR00690
  996 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF CUMULIFORM CLOUDS ICLR00700
     +NPUT EXCEEDS THE LIMIT OF 10 //)                                  CLR00710
      GO TO 999                                                         CLR00720
  997 IERR=1                                                            CLR00730
      WRITE(IOOUT,998)                                                  CLR00740
  998 FORMAT(1H0,20X,64H***CLREAD ERROR*** IMPROPER INPUT FORMAT OR ABSECLR00750
     +NT GO    SENTINEL //)                                             CLR00760
  999 RETURN                                                            CLR00770
      END                                                               CLR00780
```

```
      SUBROUTINE SCREEN(IERR,ICLMAT)
C
C  THIS MODULE   1) COMPUTES TRANSMITTANCE REQUIRED TO REDUCE THE
C                   PROBABILITY OF STATIC TARGET DETECTION BELOW A
C                   GIVEN LEVEL FOR CERTAIN TYPES OF IMAGERS. (ITAM)
C
C        - AND/OR -
C
C                2) COMPUTES HC AND WP 105 MM/155 MM SMOKE MUNITIONS
C                   REQUIRED TO PRODUCE A SMOKE SCREEN OF USER-DEFINED
C                   LENGTH AND DURATION FOR VISIBLE, NEAR, MID AND FAR
C                   IR WAVELENGTHS.
C
C   SUBROUTINES CALLED BY SCREEN ARE - CWIC AND ITAM.
C
C   ALL OUTPUT FROM SCREEN IS TABULAR.
C
C  THE PRESENT VERSION OF CWIC USES THE XSCALE MODULE TO OPTIONALLY
C     CORRECT FOR EXTINCTION DUE TO FOG, RAIN AND/OR SNOW AT IR
C     WAVELENGTHS BASED ON VISIBILITY IN THE .5 MICROMETER REGION.
C  CWIC CAN ALSO COMPUTE PASQUILL CATEGORY FROM FUNDAMENTAL MET INPUTS
C     IF THE USER CHOOSES NOT TO PROVIDE THE CATEGORY DIRECTLY.
C  CLIMATOLOGICAL VALUES FROM THE CLIMAT MODULE CAN OPTIONALLY BE
C     USED AS "TYPICAL" MET INPUTS.
C
C  THE ITAM MODULE CAN BE USED IN A LOOPING MODE THROUGH MULTIPLE SETS
C     OF INPUT RECORDS TO GENERATE TABLES.  THE LAST VALUE OF TRANS-
C     MITTANCE COMPUTED IS THAT VALUE WHICH CAN (OPTIONALLY) BE PASSED
C     TO CWIC AS THE THRESHOLD LEVEL FOR TOTAL PATH TRANSMITTANCE.
C
C  TWO RECORDS MUST BE PROVIDED TO SCREEN:
C  ONE CARD MUST BE INPUT TO SCREEN TO SELECT OPTIONS: (3(1X,I1))
C   COL 2    ICITAM  =1 CALL ITAM, OR 0 (NO CALL).
C   COL 4    ICCWIC  =1 CALL CWIC, OR 0 (NO CALL).
C   COL 6    ICCLIM  =1 USE CLIMAT FOR MET INPUTS, OR 0 USE USER VALUES.
C
C  IF CHOSEN, ALL INPUT RECORDS TO ITAM ARE READ FOLLOWING THE ABOVE
C     RECORD. (SEE INPUT DESCRIPTION IN ITAM)
C
C     AND, IF CHOSEN, INPUT RECORDS TO CWIC ARE THEN READ. (SEE INPUT
C     DESCRIPTION IN CWIC)
C
C  FINALLY, ONE RECORD IS READ BY SCREEN WITH THE WORD "END" IN COLUMNS
C     1-3.  THIS RETURNS CONTROL TO THE EOSAEL EXECUTIVE MODULE.
C
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      READ (IOIN,10) ICITAM,ICCWIC,ICCLIM
10    FORMAT(1X,I1,1X,I1,1X,I1)
      IF (ICITAM.NE.0) CALL ITAM(IERR,TFNL)
      IF (IERR.EQ.1) GOTO 15
      IF (ICCLIM.EQ.0) GOTO 20
      IF (ICLMAT.EQ.1) GOTO 20
      IERR=1
      WRITE (IOOUT,30)
30    FORMAT(1X,62H*** IN SCREEN ROUTINE, MET SOURCE SPECIFIED AS CLIMAT
     *OLOGICAL,/5X,39HBUT CLIMAT ROUTINE HAD NOT BEEN CALLED.)
      RETURN
15    WRITE (IOOUT,40)
      TFNL=1.
40    FORMAT(1X,21HIERR FLAG SET IN ITAM)
20    IF  (ICCWIC.NE.0) CALL CWIC (IERR,ICITAM,ICCLIM,TFNL)
      RETURN
      END
```

```
      SUBROUTINE CWIC (IERR,ICITAM,ICCLIM,TFNL)
C  CWIC COMPUTES THE REQUIRED SMOKE MUNITIONS EXPENDITURE (NUMBER,
C     RATE OF FIRE, PLACEMENT) TO PRODUCE A SCREEN OF DEFINED
C     LENGTH AND DURATION USING HC OR WP 105MM OR 155MM SMOKE
C     MUNITIONS.  WAVELENGTHS ARE FOR RANGES OF VISIBLE, NEAR,
C     MID AND FAR IR.
C
C  ON INPUT, IF ICITAM IS NON-ZERO, ALL COMPUTATIONS WILL USE
C     THE INPUT TFNL AS THE TOTAL SCREEN TRANSMISSION THRESHOLD.  IF
C     ICITAM IS 0, THEN BUILT-IN TOTAL THRESHOLDS OF .05 ARE USED
C     FOR ALL WAVELENGTH REGIONS.
C
C  INPUTS TO CWIC ARE ON STANDARDIZED RECORDS CONTAINING KEY-WORDS
C     IN COLUMNS 1-4 AND REAL (IE DECIMAL) VALUES FOR ALL INPUTS,
C     PLACED IN FIELDS 11-20, 21-30,..., 71-80.  KEY-WORD TYPES
C     ARE    SCRN    FOR SCREEN, OBSERVER/TARGET LOS AND ADVERSE
C                    WEATHER CORRECTIONS.
C            METR    FOR METEOROLOGICAL CONDITIONS (MAY BE OMITTED
C                        IF ICCLIM IS NON-ZERO, IN WHICH CASE THE
C                        CLIMATOLOGICAL VALUES FROM CLIMAT ARE USED.
C            PASQ    FOR (OPTIONAL) MET PARAMETERS REQUIRED TO COMPUTE
C                        THE PASQUILL STABILITY CATEGORY.  NOT REQUIRED
C                        IF PASQUILL CATEGORY ITSELF IS INPUT.
C            DONE    WHICH RETURNS EXECUTIVE CONTROL BACK TO THE SCREEN
C                        MODULE.
C
C  THE ORDER OF THE INPUT RECORDS IS IMMATERIAL, EXCEPT THAT THE (DONE)
C     CARD MUST BE LAST.
C
C  INPUTS:    (ALL VALUES REAL)   FORMAT (2A2,6X,7F10.3)
C
C  KEYWORD      COLS.      VARIABLE               DESCRIPTION
C  -------     -------     --------    ------------------------------------
C    SCRN        1-4                   SCREEN/LOS DEFINITION (REQUIRED)
C               11-20       TIME       -   SCREEN DURATION (MINUTES)
C               21-30       X0         -   SCREEN LENGTH (METERS)
C               31-40       H3         -   SLANT RANGE OBS-TARGET (KM)
C               41-50       AST        -   ELEVATION ANGLE OF TARGET FROM
C                                          OBSERVER WRT HORIZONTAL (DEG.)
C               51-60       DLS        -   COMPASS DIRECTION (CLOCKWISE WRT
C                                          NORTH) OF LINE-OF-SIGHT (DEG.)
C               61-70       ARE        -   TERRAIN ROUGHNESS ELEMENT (CM)
C               71-80       FOG        -   ADVERSE WEATHER/HAZE SELECTION
C                                            0. = NO ADVERSE WEATHER
C                                            1. = ONLY CORRECT VISIBLE WAVE-
C                                                 LENGTHS FOR INPUT VISIBILITY.
C                                            2. = CORRECT FOR FOG/HAZE FOR
C                                                   MARITIME ARCTIC AIR MASS
C                                            3. = CORRECT FOR FOG/HAZE FOR
C                                                   MARITIME POLAR AIR MASS
C                                            4. = CORRECT FOR FOG/HAZE FOR
C                                                   CONTINENTAL POLAR AIR MASS
C                                            5. = CORRECT FOR RAIN.
C                                            6. = CORRECT FOR SNOW.
C
C    METR         1-4                  MET INPUTS (NOT REQUIRED IF ICCLIM
C                                      NON-ZERO FROM SCREEN MODULE.)
C               11-20       S3         -   WINDSPEED (METERS/SEC)
C               21-30       D0         -   WIND DIRECTION (DEG) CLOCKWISE
C                                          WRT NORTH, USUAL MET CONVENTION
C               31-40       PCAT       -   PASQUILL CATEGORY. IF INPUT AS 0.
C                                          SEE PASQ RECORD BELOW.  OTHERWISE,
C                                          1.=A, 2.=B, 3.=C, 4.=D, 5.=E, 6.=F
C               41-50       VS         -   VISIBILITY (KM) NOT REQUIRED IF
C                                          FOG = 0. IS SPECIFIED.
C               51-60       R0         -   RELATIVE HUMIDITY (PERCENT). IF 0.,
C                                          THEN DEW POINT AND TEMPERATURE ARE
C                                          REQUIRED BELOW TO COMPUTE R0.
C               61-70       T0         -   AIR TEMP. (DEG C) REQUIRED IF R0 NO
C                                          GIVEN.
```

```
C               71-80       T1          -   DEW POINT TEMP. (DEG C) REQUIRED IF
C                                           R0 NOT GIVEN.
C
C     PASQ       1-4                    PASQUILL CATEGORY DETERMINATION (REQUI
C                                       RED ONLY IF PCAT ABOVE IS 0.)
C               11-20       SLAT        -   LATITUDE OF SITE (DEG.). POSITIVE I
C                                           NORTH LATITUDE.
C               21-30       SLONG       -   LONGITUDE OF SITE (DEG.). POSITIVE
C                                           EAST LONGITUDE.
C               31-40       SJDATE      -   JULIAN DATE. (DECIMAL DAYS)
C               41-50       SZHOUR      -   GMT TIME OF DAY (DECIMAL HOURS AND
C                                           FRACTIONAL HOURS)
C               51-60       C0          -   CEILING CLOUD HEIGHT (METERS)
C               61-70       C1          -   CLOUD COVER (PERCENT)
C
C     DONE       1-4                    END OF CWIC EXECUTION (REQUIRED)
C
C    ALL OUTPUT FROM CWIC IS TABULAR.  SUBROUTINES CALLED ARE -
C        CWIC1, CWIC3, CWIC4, JPASCT, XSCALE.
C
      DIMENSION CS(4,2),TRSH(4),WAVE(4),ISLANT(4),IR(8),
     *H(2,2),U(2,2),LNGTH(16),ISMOKE(2),IGUN(4),JANS(4)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN, LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /GEOMET/PTS(15),IGEOSW
      COMMON /CLYMAT/ TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,
     *WNDVEL,WNDDIR,IPASCT
      COMMON /M06/ P(4,2,2),Q1(4,2,2),Q2(4,2,2),Y(4,2,2),Z(4,2,2),
     *             C(4,2),R(4,2),V(2),RV(7),T(4),ARE,C0,C1,C2,D0,D2,
     *             DLS,R0,R2,S0,S3,SJDATE,SLAT,SLONG,SZHOUR,TIME,
     *             TR(4),X0,Y1,IT(2),IP0,IDUMMY
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      EXTERNAL JPASCT
      DATA IR/2HSC,2HRN,2HME,2HTR,2HPA,2HSQ,2HDO,2HNE/
      DATA H /18.7,77.1,1737.3,7076.2/
      DATA U /5.4,7.9,1.8,2.6/
      DATA LNGTH /2HVI,2HSI,2HBL,2HE:,2HNE,2HAR,2H I,2HR:,2HMI,
     *2HD ,2HIR,2H: ,2HFA,2HR ,2HIR,2H: /
      DATA JANS /2HNO,2H  ,2HYE,2HS /
      DATA ISMOKE /2HHC,2HWP/
      DATA IGUN /2H10,2H5 ,2H15,2H5 /
C *** INITIALIZE VALUES FOR DEFAULTS.
C     EXTINCTION COEF HC (VIS.,NEAR,MID,FAR), WP (VIS.,NEAR,MID,FAR IR)
      DATA CS /3.3,1.5,0.53,0.13,2.46,2.0,.25,.32/
C     BUILT IN TRANSMISSION THRESHOLDS, WAVELENGTHS, XSCALE SLANTS
      DATA TRSH /.05, .05, .05, .05/
      DATA WAVE /.55,1.06,3.5,10.6/,ISLANT/0,1,1,1/
      ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)
      TIME=10.
      X0=1000.
      H3=3.
      AST=0.
      DLS=90.
      ARE=1.
      FOG=0.
      S3=4.
      D0=0.
      PCAT=3.
      R0=50.
      T0=24.
      T1=12.94
      VS=10.
C READ DATA CARDS
      ICOU=0
10    READ (IOIN,900) IT(1),IT(2),(RV(I),I=1,7)
      ICOU=ICOU+1
      IF (IT(1).EQ.IR(1).AND.IT(2).EQ.IR(2)) GOTO 20
      IF (IT(1).EQ.IR(3).AND.IT(2).EQ.IR(4)) GOTO 40
      IF (IT(1).EQ.IR(5).AND.IT(2).EQ.IR(6)) GOTO 50
      IF (IT(1).EQ.IR(7).AND.IT(2).EQ.IR(8)) GOTO 60
      WRITE (IOOUT,901) IT(1),IT(2),(RV(I),I=1,7)
```

```
900   FORMAT(2A2,6X,7F10.3)                                             CWC01400
901   FORMAT(1X,51HIN CWIC, THE FOLLOWING CARD DOES NOT CONFORM TO PRO, CWC01410
     *15HPER CONVENTIONS/1X,2A2,6X,7F10.3)                              CWC01420
      IF (ICOU.LE.4) GOTO 10                                            CWC01430
      WRITE (IOOUT,902)                                                 CWC01440
902   FORMAT(1X,39HINVALID INPUTS TO CWIC. IERR=1 RETURNED)             CWC01450
      IERR=1                                                            CWC01460
      RETURN                                                            CWC01470
20    TIME=RV(1)                                                        CWC01480
      X0=RV(2)                                                          CWC01490
      H3=RV(3)                                                          CWC01500
      AST=RV(4)                                                         CWC01510
      DLS=RV(5)                                                         CWC01520
      ARE=RV(6)                                                         CWC01530
      FOG=RV(7)                                                         CWC01540
      IF(IGEOSW.NE.1) GO TO 22                                          CWC01550
      DELX=PTS(1)-PTS(4)                                                CWC01560
      DELY=PTS(2)-PTS(5)                                                CWC01570
      DELZ=PTS(3)-PTS(6)                                                CWC01580
      H3=SQRT(DELX**2+DELY**2+DELZ**2)                                  CWC01590
      HDIS=SQRT(DELX**2+DELY**2)                                        CWC01600
      RTDCON=57.29577951                                                CWC01610
      AST=RTDCON*ACOS(HDIS/H3)                                          CWC01620
      IF(HDIS.GT.1.E-20) DLS=RTDCON*ACOS(DELY/HDIS)                     CWC01630
      IF(DELX.LT.0.) DLS=360.-DLS                                       CWC01640
   22 CONTINUE                                                          CWC01650
      GOTO 10                                                           CWC01660
40    S3=RV(1)                                                          CWC01670
      D0=RV(2)                                                          CWC01680
      PCAT=RV(3)                                                        CWC01690
      VS=RV(4)                                                          CWC01700
      R0=RV(5)                                                          CWC01710
      T0=RV(6)                                                          CWC01720
      T1=RV(7)                                                          CWC01730
      GOTO 10                                                           CWC01740
50    SLAT=RV(1)                                                        CWC01750
      SLONG=RV(2)                                                       CWC01760
      SJDATE=RV(3)                                                      CWC01770
      SZHOUR=RV(4)                                                      CWC01780
      C0=RV(5)                                                          CWC01790
      C1=RV(6)                                                          CWC01800
      GOTO 10                                                           CWC01810
C***  BEGIN COMPUTATIONS.  -- FIRST MET VALUES.                         CWC01820
60    IF (ICCLIM.EQ.0) GOTO 70                                          CWC01830
C*** USE CLIMAT PASSED VALUES:                                          CWC01840
      T0=TEMP                                                           CWC01850
      T1=DP                                                             CWC01860
      R0=RH                                                             CWC01870
      PCAT=FLOAT(IPASCT)                                                CWC01880
      VS=VIS                                                            CWC01890
      D0=WNDDIR                                                         CWC01900
      S3=WNDVEL                                                         CWC01910
C*** PROVIDE WINDSPEED IN KNOTS AS S0:                                  CWC01920
70    S0=S3/.515                                                        CWC01930
      IF (S0.LE.1.) S3=.515                                             CWC01940
      IF (S0.LE.1.) S0=1.                                               CWC01950
C*** NOW CHECK RH AND COMPUTE IF NECESSARY.                             CWC01960
      IRNOT=0                                                           CWC01970
      IF (R0.GT.0.) GOTO 80                                             CWC01980
      IF(T0 .GT. 0.) GO TO 76                                           CWC01990
      A0=9.5                                                            CWC02000
      B0=265.5                                                          CWC02010
      IF(T0 .LE. 0.) GO TO 78                                           CWC02020
76    CONTINUE                                                          CWC02030
      A0=7.5                                                            CWC02040
      B0=237.3                                                          CWC02050
78    CONTINUE                                                          CWC02060
      IF(T1 .GE. 0.) GO TO 79                                           CWC02070
      A1=9.5                                                            CWC02080
      B1=265.5                                                          CWC02090
```

```
      IF(T1 .LE. 0.) GO TO 75
79    CONTINUE
      A1=7.5
      B1=237.3
75    CONTINUE
      E0=6.11*10.**((A0*T0)/(B0+T0))
      E1=6.11*10.**((A1*T1)/(B1+T1))
      R0=(E1/E0)*100.
      IRNOT=1
C*** NOW CHECK PASQUILL CATEGORY..
80    IP0=IFIX(PCAT+.001)
      IF (IP0.GT.6) IP0=6
      IPNOT=0
      IF (IP0.GT.0) GOTO 90
      CALL CWIC1 (IP0,C0,C1,SLAT,SLONG,SJDATE,SZHOUR,S0)
      IPNOT=1
C*** NEXT COMPUTE CL VALUE FOR THRESHOLD TRANSMITTANCE, CORRECTED FOR
C    FOG,RAIN,...
C*** VISIBILITY EXTINCTION (KM-1) .. NOTE THAT IF VS=0., THEN SET TO
C                           CLEAR DAY AND COMPUTATIONS CONTINUE.
90    EX55=0.
      IF (VS.GT.0.) EX55=3.912/VS
C*** TRANSMISSION THRESHOLDS..
      DO 92 I=1,4
      T(I)=TRSH(I)
      IF (ICITAM.NE.0) T(I)=TFNL
      TR(I)=T(I)
92    CONTINUE
C***  CORRECTIONS FOR WEATHER
      IFOG=-1+IFIX(FOG+.0001)
      IF (IFOG.LE.-1) GOTO 100
C*** CORRECT VISIBLE FOR VISIBILITY (H3 SLANT RNG, AST ELEV. ANG.)
      XSTRN=0.
      EPTH=EX55*H3
      IF (EPTH.LT.12.) XSTRN=EXP(-EPTH)
C SET VSET FOR NO CARD I/O IN XSCALE
      VSET=89.
      IF((AST.GT.0.1 .OR. AST.LT.-0.1).AND.IFOG.LT.5)
     *CALL XSCALE(WAVE(1),VSET,EX55,XSTRN,IERR,ISLANT(1),IFOG,H3,AST)
      IF (XSTRN.LE.0.) XSTRN=.0001
      IF (XSTRN.GT.1.) XSTRN=1.
      T(1)=T(1)/XSTRN
      IF (IFOG.EQ.0) GOTO 100
C CORRECT NON-VISIBLE FOR SEEABILITY.
94    CONTINUE
      DO 96 I=2,4
      ISLNT=ISLANT(I)
      IF (AST.EQ.0.) ISLNT=0
      IF (IFOG.GE.5) ISLNT=0
      CALL XSCALE(WAVE(I),VSET,EX55,XSTRN,IERR,ISLNT,IFOG,H3,AST)
      IF (XSTRN.LE.0.) XSTRN=.0001
      IF (XSTRN.GT.1.) XSTRN=1.
      T(I)=T(I)/XSTRN
96    CONTINUE
C*** COMPUTE CL FOR THRESHOLD TRANSMISSION REQUIRED OF SMOKE.
100   DO 108 I=1,4
      IF (T(I).GT.1.) T(I)=1.
      IF (T(I).LE.0.) T(I)=.00001
      DO 105 J=1,2
      C(I,J)=ALOG(T(I))/(-CS(I,J))
105   CONTINUE
108   CONTINUE
C*** ATMOSPHERIC DIFFUSION
      CALL CWIC3 (ARE,DLS,C2,D0,D2,H,IP0,R0,R2,U,V,Y1)
C*** MUNITIONS EXPENDITURES
      CALL CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0)
C** END MAIN CWIC COMPUTATIONS,  FINAL OUTPUT.
      IF(ICITAM.GT.0) WRITE(IOOUT,10000)
      WRITE (IOOUT,10200)
      WRITE(IOOUT,10800)
```

```
      WRITE (IOOUT,10140)
      WRITE(IOOUT,10300)
      WRITE(IOOUT,10400)
      WRITE(IOOUT,10200)
      WRITE (IOOUT,14000) H3
      WRITE (IOOUT,14100) AST
      WRITE (IOOUT,14200) DLS
      WRITE (IOOUT,11900) ARE
      WRITE (IOOUT,11400)
      IAD=1
      IF (IFOG.GE.0) IAD=3
      WRITE (IOOUT,14300) JANS(IAD),JANS(IAD+1)
      IAD=1
      IF (IFOG.EQ.1) IAD=3
      WRITE (IOOUT,14400) JANS(IAD),JANS(IAD+1)
      IAD=1
      IF (IFOG.EQ.2) IAD=3
      WRITE (IOOUT,14500) JANS(IAD),JANS(IAD+1)
      IAD=1
      IF (IFOG.EQ.3) IAD=3
      WRITE (IOOUT,14600) JANS(IAD),JANS(IAD+1)
      IAD=1
      IF (IFOG.EQ.4) IAD=3
      WRITE (IOOUT,14700) JANS(IAD),JANS(IAD+1)
      IAD=1
      IF (IFOG.EQ.5) IAD=3
      WRITE (IOOUT,14800) JANS(IAD),JANS(IAD+1)
      WRITE (IOOUT,15000)
      WRITE (IOOUT,11800) S3
      WRITE (IOOUT,11700) D0
      JP=JPASCT(IP0)
      WRITE (IOOUT,12000) JP
      WRITE (IOOUT,11300) VS
      WRITE (IOOUT,12100) R0
      IF (IRNOT.EQ.1) WRITE (IOOUT,11500) T0
      IF (IRNOT.EQ.1) WRITE (IOOUT,11600) T1
      IF (IPNOT.EQ.0) GOTO 120
      WRITE (IOOUT,15100)
      IF (SLAT.GE.0.) WRITE(IOOUT,10601) SLAT
      SLATI=-SLAT
      IF (SLAT.LT.0.) WRITE(IOOUT,10602) SLATI
      IF (SLONG.GE.0.) WRITE(IOOUT,10701) SLONG
      SLONGI=-SLONG
      IF (SLONG.LT.0.) WRITE(IOOUT,10702) SLONGI
      WRITE(IOOUT,10900) SJDATE
      WRITE(IOOUT,11000) SZHOUR
      WRITE(IOOUT,11100) C0
      WRITE(IOOUT,11200) C1
120   CONTINUE
      WRITE (IOOUT,15200)
      DO 150 I=1,4
      IWL=4*(I-1)
      WRITE (IOOUT,15300) (LNGTH(IWL+J),J=1,4),TR(I),T(I)
150   CONTINUE
      WRITE(IOOUT,10000)
C* PRINT MUNITION EXPENDITURES
      WRITE(IOOUT,10130)
      WRITE(IOOUT,16200) (LNGTH(J),J=1,8)
      WRITE(IOOUT,10100)
      WRITE(IOOUT,16300)
      WRITE(IOOUT,16400) X0,TIME,X0,TIME
      WRITE(IOOUT,10200)
      WRITE(IOOUT,16500) ISMOKE(1),ISMOKE(1)
      WRITE(IOOUT,10100)
      WRITE(IOOUT,16600) IGUN(1),IGUN(2),IGUN(1),IGUN(2)
      WRITE(IOOUT,10100)
      WRITE(IOOUT,16700)
      WRITE(IOOUT,16800)
      WRITE(IOOUT,16900) Q1(1,1,1),Y(1,1,1),Q1(2,1,1),Y(2,1,1)
      WRITE(IOOUT,17000) Q1(1,2,1),R(1,1),Y(1,2,1),P(1,1,1),Q1(2,2,1),
```

```
     *R(2,1),Y(2,2,1),P(2,1,1)
      WRITE(IOOUT,10200)
      WRITE(IOOUT,16600) IGUN(3),IGUN(4),IGUN(3),IGUN(4)
      WRITE(IOOUT,10100)
      WRITE(IOOUT,16700)
      WRITE(IOOUT,16800)
      WRITE(IOOUT,16900) Q1(1,1,2),Y(1,1,2),Q1(2,1,2),Y(2,1,2)
      WRITE(IOOUT,17000) Q1(1,2,2),R(1,1),Y(1,2,2),P(1,2,1),Q1(2,2,2),
     *R(2,1),Y(2,2,2),P(2,2,1)
      WRITE(IOOUT,10200)
      WRITE(IOOUT,16500) ISMOKE(2),ISMOKE(2)
      WRITE(IOOUT,10100)
      WRITE(IOOUT,16600) IGUN(1),IGUN(2),IGUN(1),IGUN(2)
      WRITE(IOOUT,10100)
      WRITE(IOOUT,16700)
      WRITE(IOOUT,16800)
      WRITE(IOOUT,16900) Q2(1,1,1),Z(1,1,1),Q2(2,1,1),Z(2,1,1)
      IF(IPO .GT. 4) R(I,2)=2.
      IF(IPO .GT. 4) R(2,2)=2.
      WRITE(IOOUT,17000) Q2(1,2,1),R(1,2),Z(1,1,1),P(1,1,2),Q2(2,2,1),
     *R(2,2),Z(2,1,1),P(2,1,2)
      WRITE(IOOUT,10200)
      WRITE(IOOUT,16600) IGUN(3),IGUN(4),IGUN(3),IGUN(4)
      WRITE(IOOUT,10100)
      WRITE(IOOUT,16700)
      WRITE(IOOUT,16800)
      WRITE(IOOUT,16900) Q2(1,1,2),Z(1,2,2),Q2(2,1,2),Z(2,2,2)
      IF(IPO .GT. 4) R(I,2)=1.
      IF(IPO .GT. 4) R(2,2)=1.
      WRITE(IOOUT,17000) Q2(1,2,2),R(1,2),Z(1,2,2),P(1,2,2),Q2(1,2,2),
     *R(2,2),Z(2,2,2),P(2,2,2)
      WRITE(IOOUT,10000)
      WRITE (IOOUT,10130)
      DO 7100 I=3,4
      IWL=4*(I-1)
      J1=IWL+1
      J2=IWL+4
      WRITE (IOOUT,12200) (LNGTH(J),J=J1,J2)
      WRITE (IOOUT,10100)
      WRITE (IOOUT,12300)
      WRITE (IOOUT,12400) X0,TIME
      WRITE (IOOUT,10200)
      WRITE(IOOUT,12500) ISMOKE(2)
      WRITE(IOOUT,10100)
      WRITE(IOOUT,13100)
      WRITE(IOOUT,13200)
      WRITE(IOOUT,13300) IGUN(1),IGUN(2),Q2(I,2,1),R(I,2),P(I,1,2)
      WRITE(IOOUT,13300) IGUN(3),IGUN(4),Q2(I,2,2),R(I,2),P(I,2,2)
      IF (I.EQ.3) WRITE(IOOUT,10140)
      IF (I.EQ.4) WRITE(IOOUT,10000)
7100  CONTINUE
      RETURN
C* FORMAT STATEMENTS.
10000 FORMAT(1H1)
10100 FORMAT(1H )
10200 FORMAT(1H0)
10130 FORMAT(///)
10140 FORMAT(////)
10300 FORMAT(55X,21HMUNITION EXPENDITURES)
10400 FORMAT(56X,19HFOR HC AND WP SMOKE)
10601 FORMAT(45X,35HLATITUDE                     - DEG      =,F7.2,6H NORTH)
10602 FORMAT(45X,35HLATITUDE                     - DEG      =,F7.2,6H SOUTH)
10701 FORMAT(45X,35HLONGITUDE                    - DEG      =,F7.2,6H EAST )
10702 FORMAT(45X,35HLONGITUDE                    - DEG      =,F7.2,6H WEST )
10800 FORMAT(1H0,47X,36(1H*)/48X,1H*,34X,1H*/48X,1H*,4X,
     *26HCWIC MUNITION EXPENDITURES,4X,1H*/48X,1H*,34X,1H*/
     *48X,36(1H*))
10900 FORMAT(45X,35HJULIAN DATE                  - DAY      =,F7.0)
11000 FORMAT(45X,35HGMT TIME                     - HOUR     =,F7.2)
11100 FORMAT(45X,35HCEILING                      - METERS   =,F7.1)
```

```
11200 FORMAT(45X,35HCLOUD COVER                - PERCENT  =,F7.1)
11300 FORMAT(45X,35HVISIBILITY                 - KM       =,F7.3)
11400 FORMAT(1H0,44X,35HATMOSPHERIC EXTINCTION CORRECTIONS )
11500 FORMAT(45X,35HTEMPERATURE                - DEG C    =,F7.1)
11600 FORMAT(45X,35HDEW POINT                  - DEG C    =,F7.1)
11700 FORMAT(45X,35HWIND DIRECTION             - DEG      =,F7.2)
11800 FORMAT(45X,35HWINDSPEED                  - M/SEC    =,F7.2)
11900 FORMAT(45X,35HAVG ROUGHNESS ELEMENT      - CM       =,F7.1)
12000 FORMAT(45X,35HPASQUILL CATEGORY          -          =,5X,A2)
12100 FORMAT(45X,35HRELATIVE HUMIDITY          - PERCENT  =,F7.1)
12200 FORMAT(60X,11(1H-)/60X,1H-,1X,4A2,1H-/60X,11(1H-))
12300 FORMAT(62X,6HLENGTH,8X,8HDURATION/62X,6HMETERS,9X,7HMINUTES)
12400 FORMAT(47X,6HSCREEN,8X,F7.0,9X,F7.2)
12500 FORMAT(59X,A2,13H SMOKE SCREEN/59X,15(1H-))
12600 FORMAT(59X,A2,A1,11HMM HOWITZER)
12700 FORMAT(47X,37HVOLLEY       GUNS RATE  SPACING ROUNDS)
12800 FORMAT(47X,30H                   /MIN   METERS)
12900 FORMAT(47X,11HINITIAL:    ,F5.0,6X,F8.0)
13000 FORMAT(47X,11HSUSTAINING:,F5.0,F5.1,F9.0,F7.0)
13100 FORMAT(47X,31H          ROUNDS/    RATE/    TOTAL)
13200 FORMAT(47X,32H        60 METERS   MINUTE  ROUNDS)
13300 FORMAT(47X,A2,A1,5HMM:    ,F5.0,5X,F4.0,3X,F7.0)
14000 FORMAT(45X,35HSLANT RANGE OBS-TGT        - KM       =,F7.3)
14100 FORMAT(45X,35HELEVATION OF TARGET        - DEG      =,F7.2)
14200 FORMAT(45X,35HAZIMUTH OF TARGET          - DEG      =,F7.2)
14300 FORMAT(45X,35HCORRECTED FOR VISIBILITY   -           ,4X,2A2)
14400 FORMAT(45X,35HMARITIME ARCTIC AIR MASS   -           ,4X,2A2)
14500 FORMAT(45X,35HMARITIME POLAR AIR MASS    -           ,4X,2A2)
14600 FORMAT(45X,35HCONTINENTAL POLAR AIR MASS -           ,4X,2A2)
14700 FORMAT(45X,35HCORRECTED FOR RAIN         -           ,4X,2A2)
14800 FORMAT(45X,35HCORRECTED FOR SNOW         -           ,4X,2A2)
15000 FORMAT(1H0,44X,35HMETEOROLOGICAL INPUTS              )
15100 FORMAT(1H0,44X,35HINPUTS FOR PASQUILL CATEGORY       )
15200 FORMAT(1H0,44X,42HTRANSMISSION THRESHOLDS    TOTAL       SMOKE)
15300 FORMAT(45X,4A2,18X,F5.3,6X,F5.3)
16200 FORMAT(33X,12(1H-),44X,12(1H-)/33X,1H-,1X,4A2,1X,1H-,44X,1H-,1X,
     *4A2,1X,1H-/33X,12(1H-),44X,12(1H-))
16300 FORMAT(1X,2(34X,6HLENGTH,8X,8HDURATION)/1X,2(34X,6HMETERS,9X,
     *7HMINUTES))
16400 FORMAT(1X,2(19X,6HSCREEN,8X,F7.0,9X,F7.2))
16500 FORMAT(32X,A2,13H SMOKE SCREEN,41X,A2,13H SMOKE SCREEN/32X,
     *15(1H-),41X,15(1H-))
16600 FORMAT(32X,A2,A1,11HMM HOWITZER,42X,A2,A1,11HMM HOWITZER)
16700 FORMAT(1X,2(19X,6HVOLLEY,6X,25HGUNS RATE  SPACING ROUNDS))
16800 FORMAT(37X,4H/MIN,3X,6HMETERS,43X,4H/MIN,3X,6HMETERS)
16900 FORMAT(20X,11HINITIAL:    ,F5.0,6X,F8.0,26X,11HINITIAL:    ,
     *F5.0,6X,F8.0)
17000 FORMAT(1X,2(19X,11HSUSTAINING:,F5.0,F5.1,F9.0,F7.0))
      END
```

```
      SUBROUTINE CWIC1 (IP0,C0,C1,SLAT,SLONG,SJDATE,SZHOUR,S0)
      DIMENSION ITAB(7,9)
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA ITAB /
     1         1,1,2,3,4,6,6,
     2         1,2,2,3,4,6,6,
     3         1,2,3,4,4,5,6,
     4         2,2,3,4,4,5,6,
     5         2,2,3,4,4,4,5,
     6         2,3,3,4,4,4,5,
     7         3,3,4,4,4,4,5,
     8         3,3,4,4,4,4,4,
     9         3,4,4,4,4,4,4/
      ASIN(ARG)=ATAN2(ARG,SQRT(1.-ARG**2))
C*METEOROLOGICAL CALCULATIONS.
C
      IF(C1 .NE. 100.) GO TO 1000
      IF(C0 .GT. 2133.6042) GO TO 1000
      I1=0
      I2=0
      GO TO 2300
1000  CONTINUE
C CALCULATE ANGULAR FRACTION OF A YEAR FOR A GIVEN JULIAN DATE.
      R9=PIRAD
      D9=1./R9
      SLAT1=SLAT*R9
      A0=((SJDATE-1.)*360.)/365.242
C CALCULATE SOLAR DECLINATION ANGLE (A4).
      A1=A0*R9
      A2=279.9348+A0
      A2=A2+(1.914827*SIN(A1))-(0.079525*COS(A1))
      A2=A2+(0.019938*SIN(2*A1))-(0.00162*COS(2*A1))
      A2=A2*R9
      A3=23.4438*R9
      A4=SIN(A3)*SIN(A2)
      A4=ASIN(A4)
C CALCULATE THE TIME OF MERIDIAN PASSAGE - TRUE SOLAR NOON (A5).
      A5= 12.+(0.12357*SIN(A1))-(0.004289*COS(A1))
      A5=A5+(0.153809*SIN(2*A1))+(0.060783*COS(2*A1))
C CALCULATE SOLAR HOUR ANGLE (A6)*** NOTE THIS VERSION USES + SIGN
C ON SLONG DUE TO EAST-LONGITUDE POSITIVE CONVENTION.
      A6=15.*(SZHOUR-A5)+SLONG
      A6=A6*R9
C CALCULATE SOLAR ALTITUDE (A7).
      A7=SIN(SLAT1)*SIN(A4)+COS(SLAT1)*COS(A4)*COS(A6)
      A7=ASIN(A7)
1100  CONTINUE
      A7=A7*D9
C CALCULATE INSOLATION CLASS NUMBER.
      I2=0
      IF(A7 .LE. 60.) GO TO 1200
      I2=4
      GO TO 1500
1200  CONTINUE
      IF(A7 .LE. 35.) GO TO 1300
      I2=3
      GO TO 1500
1300  CONTINUE
      IF(A7 .LE. 15.) GO TO 1400
      I2=2
      GO TO 1500
1400  CONTINUE
      IF(A7 .LE. 0.) GO TO 2200
      I2=1
C CALCULATE NET RADIATION INDEX FOR DAYTIME.
1500  CONTINUE
      I3=0
      IF(C1 .GT. 50.) GO TO 1600
      I3=I2
      GO TO 1900
```

```
1600  CONTINUE
      IF(C0 .GT. 2133.6042) GO TO 1700
      I3=I2-2
      GO TO 1900
1700  CONTINUE
      IF(C0 .GE. 4876.8096) GO TO 1800
      I3=I2-1
      GO TO 1900
1800  CONTINUE
      IF(C1 .NE. 100.) GO TO 1900
      I3=I2-1
1900  CONTINUE
      IF(I3 .NE. 0) GO TO 2000
      I3=I2
2000  CONTINUE
      IF(I3 .GT. 1) GO TO 2100
      I3=1
2100  CONTINUE
      I1=I3
      GO TO 2300
C COMPUTE NET RADIATION INDEX FOR NIGHTTIME.
2200  CONTINUE
      IF(C1 .GT. 40.) GO TO 2250
      I1=-2
      GO TO 2300
2250  CONTINUE
      I1=-1
C CALCULATE PASQUILL STABILITY CATAGORY.
2300  CONTINUE
      I4=0
      I5=0
      IF(I1 .NE. 4) GO TO 2400
      I4=1
2400  CONTINUE
      IF(I1 .NE. 3) GO TO 2420
      I4=2
2420  CONTINUE
      IF(I1 .NE. 2 ) GO TO 2440
      I4=3
2440  CONTINUE
      IF(I1 .NE. 1) GO TO 2460
      I4=4
2460  CONTINUE
      IF(I1 .NE. 0) GO TO 2480
      I4=5
2480  CONTINUE
      IF(I1 .NE. -1) GO TO 2500
      I4=6
2500  CONTINUE
      IF(I1 .NE. -2) GO TO 2520
      I4=7
2520  CONTINUE
      IF(S0 .GE. 2.) GO TO 2540
      I5=1
      GO TO 2700
2540  CONTINUE
      IF(S0 .GE. 4.) GO TO 2560
      I5=2
      GO TO 2700
2560  CONTINUE
      IF(S0 .GE. 6.) GO TO 2580
      I5=3
      GO TO 2700
2580  CONTINUE
      IF(S0 .GE.  7.) GO TO 2600
      I5=4
      GO TO 2700
2600  CONTINUE
      IF(S0 .GE.  8.) GO TO 2620
      I5=5
```

```
      GO TO 2700                                                        CWX01400
 2620 CONTINUE                                                          CWX01410
      IF(S0 .GE. 10.) GO TO 2640                                        CWX01420
      I5=6                                                              CWX01430
      GO TO 2700                                                        CWX01440
 2640 CONTINUE                                                          CWX01450
      IF(S0 .GE. 11.) GO TO 2660                                        CWX01460
      I5=7                                                              CWX01470
      GO TO 2700                                                        CWX01480
 2660 CONTINUE                                                          CWX01490
      IF(S0 .GE. 12.) GO TO 2680                                        CWX01500
      I5=8                                                              CWX01510
      GO TO 2700                                                        CWX01520
 2680 CONTINUE                                                          CWX01530
      I5=9                                                              CWX01540
 2700 CONTINUE                                                          CWX01550
      IP0=ITAB(I4,I5)                                                   CWX01560
      RETURN                                                            CWX01570
      END                                                               CWX01580
```

```
      SUBROUTINE CWIC3 (ARE,DLS,C2,D0,D2,H,IP0,R0,R2,U V,Y1)
      DIMENSION A(6),S(6,3),D(6,3),H(2,2),U(2,2),V(2)
      DIMENSION W(6)
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA A/0.4,0.32,0.22,0.144,0.102,0.076/
      DATA S /
     *.139085297,.122097643,.110104377,.097649832,.070772166,.055487093,
     *.015017284,.010970370,.010962963,.010418519,7.27284E-3,6.55309E-3,
     *-1.02581E-4,-6.80135E-5,-6.73401E-5,-6.83502E-5,-4.50056E-5,
     *-4.01796E-5/
      DATA D /
     *.944814815,.894803591,.854792368,.816026936,.786026936,.726015713,
     *-4.85185E-3,-4.83951E-3,-4.82716E-3,-6.07407E-3,-6.07407E-3,
     *-6.06173E-3 ,
     *3.7037E-5,3.59147E-5,3.47924E-5,4.7138E-5,4.7138E-5,4.60157E-5/
      DATA W/0.016,0.016,0.016,0.016,0.016,0.016/
C*ATMOSPHERIC DIFFUSION CALCULATIONS.
      A1=-1.24+1.19*ALOG10(ARE)
      Z1=10.**A1
      A2=ABS(DLS-D0)*(PI/180.)
      R2=SQRT(13.69/(13.69*SIN(A2)*SIN(A2)+COS(A2)*COS(A2)))
      Y1=1.09521547+(0.02906894*R0)-(4.9575E-04*R0*R0)+
     2                (4.82E-06*R0*R0*R0)
      Y2=3.364059144+(0.060502571*R0)-(1.15301E-03*R0*R0)+
     2                (1.33942E-05*R0*R0*R0)
      C2=S(IP0,1)+S(IP0,2)*Z1+S(IP0,3)*Z1**2
      D1=D(IP0,1)+D(IP0,2)*Z1+D(IP0,3)*Z1**2
      D2=1/D1
      DO 5400 I=1,4
C*CALCULATE CROSSWIND INTEGRATED CONCENTRATION FOR WP SMOKE.
      DO 5300 K=1,2
      IF (I .LT. 3 .AND. IP0 .GT. 4) GOTO 5300
      S1=U(K,1)+0.74*A(IP0)*100.**0.9
      S2=U(K,2)+0.667*C2*100.**D1
      V(K)=(W(IP0)*Y2*H(K,2))/(PI*S1*S2)
5300  CONTINUE
5400  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0)
      REAL ME(2)
      DIMENSION C(4,2),H(2,2),P(4,2,2),Q1(4,2,2),Q2(4,2,2),R(4,2),V(2),
     *Y(4,2,2),Z(4,2,2)
      DATA ME/0.4,0.4/
C*MUNITIONS EXPENDITURES.
      DO 6900 I=1,4
C* CALCULATE INITIAL SHELL SPACING FOR HC SMOKE
      DO 6800 K=1,2
      IF(I .GT. 2) GO TO 6100
      Y(I,1,K)=45.*S3
C* CALCULATE SUSTAINING SHELL SPACING FOR HC SMOKE
      IF(C(I,1) .NE. 0.) GO TO 5500
      Y(I,2,K)=0.
      GO TO 5600
5500  CONTINUE
      Y(I,2,K)=1/R2*((0.731*ME(K)*Y1*H(K,1))/(C2*S3*C(I,1)))**D2
      IF(Y(I,2,K) .GT. X0) Y(I,2,K)=X0
5600  CONTINUE
      IF(Y(I,2,K) .NE. 0.) GO TO 5700
      Q1(I,1,K)=1.
      Q1(I,2,K)=1.
      GO TO 5900
5700  CONTINUE
C* CALCULATE INITIAL VOLLEY FOR HC SMOKE
      IF(Y(I,1,K) .GT. Y(I,2,K)) Y(I,1,K)=Y(I,2,K)
      Q1(I,1,K)=X0/Y(I,1,K)
      Q5=AINT(Q1(I,1,K))
      Q6=Q1(I,1,K)-Q5
      IF(Q6 .EQ. 0.) GO TO 5800
      Q1(I,1,K)=Q5+1.
5800  CONTINUE
C* CALCULATE NUMBER OF GUNS FOR SUSTAINING VOLLEYS (HC)
      Q1(I,2,K)=X0/Y(I,2,K)
      Q5=AINT(Q1(I,2,K))
      Q6=Q1(I,2,K)-Q5
      IF(Q6 .EQ. 0.) GO TO 5900
      Q1(I,2,K)=Q5+1.
5900  CONTINUE
C* CALCULATE RATE OF FIRE FOR HC SMOKE
      R(I,1)=0.5
      IF(C(I,1) .NE. 0.) GO TO 6000
      R(I,1)=0.
6000  CONTINUE
C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (HC SMOKE)
      P(I,K,1)=Q1(I,1,K)+(0.5*Q1(I,2,K)*(TIME-2.))
      Q5=AINT(P(I,K,1))
      Q6=P(I,K,1)-Q5
      IF(Q6 .EQ. 0.) GO TO 6100
      P(I,K,1)=Q5+1.
6100  CONTINUE
      IF(I .LT. 3 .AND. IP0 .GT.4) GO TO 6775
C* SHELL SPACING (Z( )) AND VOLLEYS (Q( )) - WP SMOKE
      IF(C(I,2) .NE. 0.) GO TO 6200
      Z(I,1,K)=0.
      Z(I,2,K)=0.
      Q2(I,1,K)=0.
      Q2(I,2,K)=0.
      GO TO 6400
6200  CONTINUE
      IF(I .LT. 3) GO TO 6250
      IF (I .GT. 2) Q2(I,1,K)=0.6*C(I,2)/V(K)
      GO TO 6300
 6250 CONTINUE
      Z(I,1,K)=V(K)/C(I,2)*100.
      Z(I,2,K)=Z(I,1,K)
      Q2(I,1,K)=X0/Z(I,2,K)+1.
6300  CONTINUE
      Q5=AINT(Q2(I,1,K))
      Q6=Q2(I,1,K)-Q5
```

```
      IF (Q6 .EQ. 0.) GO TO 6350
      Q2(I,1,K)=Q5+1.
6350  CONTINUE
      Q2(I,2,K)=Q2(I,1,K)
6400  CONTINUE
C* RATE OF FIRE FOR WP SMOKE
      IF(C(I,2) .NE. 0.) GO TO 6425
      R(I,2)=0.
      GO TO 6600
6425  CONTINUE
      IF (I .GT. 2) GOTO 6450
      R(I,2)=(Z(I,2,K)+60.)/S3
      GO  TO 6475
6450  CONTINUE
      R(I,2)=120./S3
6475  CONTINUE
      R(I,2)=R(I,2)/20.
      R5=AINT(R(I,2))
      R6=R(I,2)-R5
      IF(R6 .LT. 0.5) GO TO 6500
      R5=R5+1.
6500  CONTINUE
      IF(R5 .NE. 0.) GO TO 6550
      R5=1.
6550  CONTINUE
      R(I,2)=R5*20./60.
6600  CONTINUE
C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (WP)
      IF(C(I,2) .NE. 0.) GO TO 6650
      P(I,K,2)=0.
      GO TO 6700
6650  CONTINUE
      IF(I .GT. 2) GO TO 6700
      P(I,K,2)=Q2(I,2,K)*(1./R(I,2))*(TIME-R(I,2))
      GO TO 6750
6700  CONTINUE
      P(I,K,2)=Q2(I,1,K)*(X0/60.+1.)*(1./R(I,2))*(TIME-R(I,2))
6750  CONTINUE
      Q5=AINT(P(I,K,2))
      Q6=P(I,K,2)-Q5
      IF(Q6 .EQ. 0.) GO TO 6800
      P(I,K,2)=Q5+1.
      GO TO 6800
6775  CONTINUE
C* CALCULATIONS FOR E AND F STABILIY CAT (STABLE FLOW)
C* INITIAL SHELL SPACING FOR WP SMOKE
      IF(I .EQ. 1 .AND. K .EQ. 1) Z(I,1,K)=100.
      IF(I .EQ. 2 .AND. K .EQ. 1) Z(I,1,K)=50.
      IF(I .LT. 3 .AND. K .EQ. 2) Z(I,1,K)=100.
C* SUSTAINING SHELL SPACING FOR WP SMOKE
      IF(I .EQ. 1 .AND. K .EQ. 1) Z(I,2,K)=100.
      IF(I .EQ. 1 .AND. K .EQ. 2) Z(I,2,K)=200.
      IF(I .EQ. 2 .AND. K .EQ. 1) Z(I,2,K)=50.
      IF(I .EQ. 2 .AND. K .EQ. 2) Z(I,2,K)=100.
C* INITIAL VOLLEY FOR WP SMOKE
      Q2(I,1,K)=X0/Z(I,1,K)+1.
C* SUSTAINING VOLLEY FOR WP SMOKE
      Q2(I,2,K)=X0/Z(I,2,K)+1.
C* RATE OF FIRE FOR WP SMOKE
      IF(K .EQ. 1) R(I,2)=.5
      IF(K .EQ. 2) R(I,2)=1.
C* TOTAL NUMBER OF ROUNDS REQUIRED (WP)
      P(I,K,2)=Q2(I,1,K)+Q2(I,2,K)*1./R(I,2)*(TIME-R(I,2))
6800  CONTINUE
      R(I,2)=1./R(I,2)
6900  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITAM(IERR,TFNL)
C
C    THIS ROUTINE IS AN INVERSION OF THE NV&EOL TARGET ACQUISITION
C       MODEL, WITH EMPHASIS ON THE DEGRADATION (TRANSMITTANCE)
C       REQUIRED TO PREVENT DETECTION ABOVE A GIVEN PROBABILITY.
C
C
C    INPUTS ARE ON STANDARDIZED RECORDS:
C
C       KEY WORD - COLUMNS 1-4, FROM AMONG TARV,SENS,GO,PAGE AND DONE.
C       DATA FIELDS, ALL REAL - COLS 11-20, 21-30, 31-40,...,71-80.
C
C       INPUT CARDS ARE ORDER INDEPENDENT, WITH ARBITRARY OR SYSTEM
C       DEFAULTS CHOSEN IF CARDS ARE LEFT OUT, OR IF VALUES ARE INPUT 0.
C       THE EXCEPTION TO ORDER INDEPENDENCE IS THAT AFTER EACH SET OF
C       INPUTS A (GO) CARD MUST BE PLACED TO INITIATE EXECUTION OF
C       ONE LOOP THROUGH THE PROGRAM.  FOLLOWING, OR IN PLACE OF
C       THE LAST INPUT-SET GO CARD, A (DONE) CARD MUST BE PROVIDED TO
C       CAUSE CONTROL TO EXIT ITAM AND RETURN TO THE SCREEN EXEC MODULE.
C
C       IN SUBSEQUENT RUNS (DELINEATED BY GO CARDS) TABLES MAY BE
C       PRODUCED LINE-BY-LINE, AND ANY INPUT WHICH HAS NOT
C       BEEN CHANGED (IE INPUT AS 0.) WILL USE THE VALUE GIVEN ON THE
C       PREVIOUS RUN. EXCEPTIONS ARE THE FOV AND AMAG VALUES WHICH MUST
C       BE SPECIFIED WHENEVER A NEW DEVICE NUMBER (LSC) IS INPUT, OR
C       THESE VALUES ASSUME DEFAULTS FOR THE DEVICE.
C
C    KEY WORD     COLS.     VARIABLE              DESCRIPTION
C    --------    -------    --------    ----------------------------------
C      TARV        1-4                  TARGET/SCENARIO DESCRIPTION
C                 11-20        ACON     -  INTRINSIC CONTRAST (DIMENSION-
C                                          LESS) OR TEMPERATURE DIFFERENCE
C                                          OF TARGET/BACKGROUND (DEG K)
C                                          FOR THERMAL DEVICES.
C                 21-30        SOG      -  SKY/GROUND RATIO, (DIMENSION-
C                                          LESS), USED FOR NON-THERMAL
C                                          DEVICES. SOG SHOULD INCLUDE
C                                          A FACTOR FOR CLOUD REFLECTANCE
C                 31-40       R         -  RANGE TO TARGET (KM)
C                 41-50       TARSZ     -  TARGET MINIMUM DIMENSION (M).
C                 51-60       ZONE      -  SEARCH ZONE (DEG.**2)
C                 61-70       ALFLG     -  AMBIENT ILLUM. CATEGORY (SEE
C                                          LIST BELOW) CONVERTED TO NAL,
C                                          AN INTEGER, IN PROGRAM. USED
C                                          FOR NON-THERMAL DEVICES.
C                 71-80       ALIGHT    -  IF ALFLG=0., THE USER MUST
C                                          PROVIDE AN AMBIENT ILLUM. HERE
C                                          (FT. CDLS.)
C
C      SENS        1-4                  SENSOR DESCRIPTION
C                 11-20       PS        -  INPUT PROBABILITY OF DETECTION.
C                                          (DIMENSIONLESS)
C                 21-30       DVNUM     -  DEVICE NUMBER. CONVERTED TO LSC,
C                                          AN INTEGER, IN PROGRAM.
C                 31-40       DMODE     -  OPERATIONAL MODE (1. = WIDE FOV,
C                                          2. = NARROW FOV). CONVERTED TO
C                                          MODE, AN INTEGER, IN PROGRAM.
C                 41-50       FOV       -  FIELD OF VIEW (DEG.)
C                 51-60       AMAG      -  MAGNIFICATION (FOR VISIBLE ONLY)
C                 61-70       AJOB      -  LEVEL OF ACQUISITION (MEDIAN
C                                          RESOLVABLE NORMALIZED CYCLES,
C                                          USUALLY 1. FOR DETECTION)
C
C      PAGE        1-4                  (OPTIONAL) - FORCE PAGE EJECT,
C                                       WRITE NEW HEADER. (USEFUL FOR TABLE
C                                       GENERATION.)
C
C      GO          1-4                  EXECUTE ONE LOOP WITH GIVEN INPUTS
C
C      DONE        1-4                  END COMPUTATIONS AND EXIT THE ITAM
```

```
C                                              ROUTINE. (MAY BE USED IN PLACE OF
C                                              OR FOLLOWING THE LAST GO CARD)
C
C     OUTPUTS:  TFNL, THE TOTAL PATH TRANSMITTANCE (DIMENSIONLESS)
C                     REQUIRED TO KEEP THE PROBABILITY OF DETECTION LESS
C                     THAN OR EQUAL TO THAT INPUT, AS COMPUTED FROM THE
C                     FINAL SET OF INPUTS PRECEDING THE DONE CARD.
C
C     PRINT OUTPUTS: ALL INPUT PARAMETERS AND
C            C      = CONTRAST OR TEMPERATURE DIFFERENCE AT DEVICE
C                     REQUIRED FOR GIVEN LEVEL OF DETECTION PROBABILITY
C            RC     = RESOLVABLE CYCLES REQUIRED AT TARGET FOR GIVEN
C                     PROBABILITY OF DETECTION.
C            TTOT   = TOTAL PATH TRANSMITTANCE TO REDUCE ACON AT TARGET
C                     TO C AT DEVICE.
C
C    SUBROUTINES:  ITAM CALLED BY SCREEN. ITAM CALLS CYCLE, CINV, TREQ.
C
C
C     AMBIENT ILLUMINATION CATEGORIES:
C
C            ALFLG OR NAL      FT. CDLS.         APPROX. CONDITIONS
C            ------------      ----------        -------------------
C                 0            USER SPEC.
C                 1            100.              CLEAR TO SLIGHT OVERCAST
C                 2             10.              HEAVILY OVERCAST DAY
C                 3              1.              SUNSET
C                 4              0.1             1/4 HR. AFTER SUNSET
C                 5              0.01            1/2 HR. AFTER SUNSET
C                 6              0.001           MOONLIT, CLEAR
C                 7              0.0001          MOONLIT, OVERCAST
C                 8              0.00001         STARLIGHT ONLY
C                 9              0.000001        MINIMAL STARLIGHT
C
      DIMENSION RV(7),IT(2),AMAGW(14),AMAGN(14),FOVW(14),FOVN(14),
     *ALFA(9),IR(10),BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /GEOMET/PTS(15),IGEOSW
      DATA IR/2HDO,2HNE,2HGO,2H  ,2HTA,2HRV,2HSE,2HNS,2HPA,2HGE/
      DATA AMAGW /1.0,7.0,12*0./,
     *     AMAGN /1.0,7.0,12*0./,
     *     FOVW /24.5,8.0,15.0,9.0,6.0,4.67,4.67,10.62,9.24,8.0,4*0./,
     *     FOVN /24.5,8.0,3*0.,1.56,0.,3.54,2.31,1.0,4*0./
      DATA TGT,SGR,FV,RN,ALV,DMG,AJB,PD,DVN,DMD,ACN /2.3,
     *     3.,7.,1.,1.,1.,1.,.1,2.,1.,1./
      DATA ALFA /1.089005,.009730,1.8528,-0.0002197,.139,0.,0.,0.,0./
      DATA BETA/-1.801654,.03536,-3.13169,.0514,0.,.0651,.1124,.0968,
     *.12/
      DATA GAMA/4.217775,2.3575,5.17227,2.32178,2.7933,1.6149,1.3364,
     *.56867,.29/
      DATA AMRAL/.05850156,.1022697,.2444688,.7264854,3.736508,
     *-16.87941, .06709132,.1137755,.3177794,.7642010,6.549327,
     *-16.18440, .04109820,.0715830,.1524969,.2319469,1.494814,
     *5.731956/
      DATA AMRGM/2.94273,2.43789,2.11113,2.05233,3.33957,-4.98529,
     *           5.22111,4.28211,4.13668,3.44685,9.16003,-7.70891,
     *           2.80725,2.70329,2.48447,1.63044,2.73650,3.47838 /
C ***** INITIALIZE VALUES (FIRST EXECUTION)
      PB=.632
      ACON=0.
      SOG=0.
      RNG=0.
      TARSZ=0.
      UZONE=0.
      ALFLG=0.
      ALIGHT=0.
      PDET=0.
      DVNUM=0.
      DMODE=0.
      DFOV=0.
```

```
      DMAG=0.
      AJOB=0.
      TBAR=0.
      NAL=0
      TFNL=1.
      ILAST=0
      IFLGFV=0
      IFLGMG=0
C ***** OUTPUT LINE COUNTER
      ILINE=0
C ***** INITIALIZE VALUES (REPEATED EXECUTIONS)
C ***** INPUT CARD COUNTER
4     ICOV=0
      IF (ILAST.NE.0) GOTO 6
C ***** FLAG TO WRITE TITLE
      ITITL=0
      TTOT=1.
      RC=0.
      C=0.
      ICM=0
      INEWD=0
C ***** READ A CARD
5     READ(IOIN,900) IT(1),IT(2),(RV(I),I=1,7)
900   FORMAT(2A2,6X,7F10.2)
      ICOV=ICOV+1
      IF(IT(1).EQ.IR(1).AND.IT(2).EQ.IR(2)) GO TO 6
      IF(IT(1).EQ.IR(3).AND.IT(2).EQ.IR(4)) GO TO 9
      IF(IT(1).EQ.IR(5).AND.IT(2).EQ.IR(6)) GO TO 7
      IF(IT(1).EQ.IR(7).AND.IT(2).EQ.IR(8)) GO TO 8
      IF (IT(1).EQ.IR(9).AND.IT(2).EQ.IR(10)) GOTO 2
      WRITE(IOOUT,901) IT(1),IT(2),(RV(I),I=1,7)
901   FORMAT(1X,61HTHE FOLLOWING CARD DOES NOT CONFORM TO ITAM INPUT CON
     *VENTIONS/1X,2A2,6X,7E10.3)
      IF(ICOV.LE.5) GO TO 5
      IERR=1
      GOTO 1
C ***** ALL DONE
6     IF (ICOV.GT.1) GOTO 3
1     WRITE(IOOUT,902) TFNL
902   FORMAT(1H0,5X,41H*** FINAL TOTAL TRANSMISSION FROM ITAM = ,F5.3)
      RETURN
C ***** TARGET CARD PROCESSING (TARV)
7     RCHK=ACON
      IF(RV(1).NE.0.) ACON=RV(1)
      IF(ACON.NE.RCHK) ITITL=1
      IF(RV(2).NE.0.) SOG=RV(2)
      IF(RV(3).NE.0.) RNG=RV(3)
      IF(IGEOSW.NE.1) GO TO 477
      RNG=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+(PTS(3)-PTS(6))**2)
  477 CONTINUE
      RCHK=TARSZ
      IF(RV(4).NE.0.) TARSZ=RV(4)
      IF(TARSZ.NE.RCHK) ITITL=1
      RCHK=UZONE
      IF(RV(5).NE.0.) UZONE=RV(5)
      IF(RCHK.NE.UZONE) ITITL=1
      IF(RV(6).NE.0.) ALFLG=RV(6)
      IF(RV(7).NE.0.) ALIGHT=RV(7)
      GO TO 5
C ***** SENSOR CARD PROCESSING (SENS)
8     IF(RV(1).GT.0.) PDET=RV(1)
      RCHK=DVNUM
      IF(RV(2).NE.0.) DVNUM=RV(2)
      IF(DVNUM.NE.RCHK) ITITL=1
      IF (DVNUM.NE.RCHK) INEWD=1
      RCHK=DMODE
      IF(RV(3).NE.0.) DMODE=RV(3)
      IF(DMODE.NE.RCHK) ITITL=1
      IF (INEWD.EQ.1) DFOV=0.
      IF (INEWD.EQ.1) IFLGFV=0
```

```
      RCHK=DFOV
      IF(RV(4).NE.0.) DFOV=RV(4)
      IF (RV(4).NE.0..AND.IFLGFV.EQ.1) IFLGFV=0
      IF(DFOV.NE.RCHK) ITITL=1
      IF (INEWD.EQ.1) DMAG=0.
      IF (INEWD.EQ.1) IFLGMG=0
      RCHK=DMAG
      IF(RV(5).NE.0.) DMAG=RV(5)
      IF (RV(5).NE.0..AND.IFLGMG.EQ.1) IFLGMG=0
      IF(DMAG.NE.RCHK) ITITL=1
      RCHK=AJOB
      IF(RV(6).NE.0.) AJOB=RV(6)
      IF(AJOB.NE.RCHK) ITITL=1
      GO TO 5
C ***** PAGE TO NEW HEADER
2     ITITL=1
      GOTO 5
C ***** BEGIN PROCESSING (GO)
C ***** FIRST CHECK IF DEFAULTS NEEDED, AND SET INTEGER VALUES, LIGHT CA
3     ILAST=1
9     RCHK=DVNUM
      IF(DVNUM.EQ.0.) DVNUM=DVN
      IF(RCHK.NE.DVNUM) ITITL=1
      LSC=IFIX(DVNUM+.0001)
      IF (LSC.LT.1) LSC=1
      IF (LSC.GT.14) LSC=14
      RCHK=DMODE
      IF(DMODE.EQ.0.) DMODE=DMD
      IF(RCHK.NE.DMODE) ITITL=1
      MODE=IFIX(DMODE+0.0001)
      IF(MODE.LT.1) MODE=1
      IF(MODE.GT.2) MODE=2
      IF(PDET.EQ.0.) PDET=PD
      PS=PDET
      IF(PS.GT.1.) PS=1.
      IF(PS.LT.0.) PS=0.
      RCHK=DFOV
      IF(DFOV.LE.0.0.AND.MODE.EQ.1) DFOV=FOVW(LSC)
      IF(DFOV.LE.0.0.AND.MODE.EQ.2) DFOV=FOVN(LSC)
      IF(DFOV.LE.0.0) IFLGFV=1
      IF(DFOV.LE.0..AND.MODE.EQ.1) DFOV=FV
      IF(DFOV.LE.0..AND.MODE.EQ.2) DFOV=FV/2.
      IF(RCHK.NE.DFOV)  ITITL=1
      FOV=DFOV
      RCHK=DMAG
      IF(DMAG.LE.0.0.AND.MODE.EQ.1) DMAG=AMAGW(LSC)
      IF(DMAG.LE.0.0.AND.MODE.EQ.2) DMAG=AMAGN(LSC)
      IF(DMAG.LE.0.0) IFLGMG=1
      IF(DMAG.LE.0.0) DMAG=DMG
      IF(RCHK.NE.DMAG) ITITL=1
      AMAG=DMAG
      RCHK=AJOB
      IF(AJOB.EQ.0.0) AJOB=AJB
      IF(RCHK.NE.AJOB) ITITL=1
      RCHK=ACON
      IF(ACON.EQ.0.0) ACON=ACN
      IF(RCHK.NE.ACON) ITITL=1
      C=ACON
      IF(SOG.EQ.0.0) SOG=SGR
      IF(RNG.LE.0.0) RNG=RN
      R=RNG
      RCHK=TARSZ
      IF(TARSZ.LE.0.0) TARSZ=TGT
      IF(RCHK.NE.TARSZ) ITITL=1
      DIM=TARSZ
      RCHK=UZONE
      IF(UZONE.LE.0.0) UZONE=FOV**2
      IF(RCHK.NE.UZONE) ITITL=1
      ZONE=UZONE
      IF (ALFLG.LT.0.) ALFLG=0.
```

```
      IF (ALFLG.GT.9.) ALFLG=9.
      IAL=IFIX(ALFLG+0.0001)
      IF(IAL.GT.0) AL=10.**(3-IAL)+1.E-10
      IF(IAL.EQ.0.AND.ALIGHT.EQ.0.0) ALIGHT=ALV
      IF(IAL.EQ.0) AL=ALIGHT
C ***** CHECK AND WRITE PAGE TITLE HEADING
10    ILINE=ILINE+1
      IF(ITITL.EQ.0.AND.ILINE.LE.48) GO TO 11
      IF (ITITL.EQ.0.AND.ILINE.GT.49) WRITE (IOOUT,905)
905   FORMAT(1H ,35X,28H** CONTINUED ON NEXT PAGE **)
      WRITE(IOOUT,910)
910   FORMAT(1H0/     41X,43(1H*)/41X,1H*,41X,1H*/41X,1H*,2X,37HINVERSE S
     *TATIC TARGET DETECTION MODEL,2X,1H*/41X,1H*,41X,1H*/41X,43(1H*))
      WRITE(IOOUT,911)
911   FORMAT(1H0,64X,28HTARGET INTRINSIC CONTRAST OR)
      WRITE(IOOUT,912) LSC,ACON
912   FORMAT(20X,13HDEVICE NUMBER,4X,I2,32X,22HTEMPERATURE DIFFERENCE,
     *      2X,F7.3)
      IF(MODE.EQ.2) WRITE(IOOUT,913) DIM
913   FORMAT(1H0,19X,11HFOV TYPE - ,4X,6HNARROW,24X,28HMINIMUM TARGET DI
     *MENSION (M),2X,F7.3)
      IF(MODE.EQ.1)WRITE(IOOUT,914)DIM
914   FORMAT(1H0,19X,11HFOV TYPE - ,6X,4HWIDE,24X,28HMINIMUM TARGET DIME
     *NSION (M),2X,F7.3)
      IF(IFLGFV.EQ.0) WRITE(IOOUT,915) FOV,AJOB
915   FORMAT(1H0,19X,9HFOV (DEG),5X,F7.3,24X,27HACQUISITION LEVEL (50 PC
     *NT),3X,F7.3)
      IF(IFLGFV.EQ.1) WRITE(IOOUT,916) FOV,AJOB
916   FORMAT(1H0,19X,9HFOV (DEG),5X,F7.3,1X,19H(ARBITRARY DEFAULT),
     *      4X,27HACQUISITION LEVEL (50 PCNT),3X,F7.3)
      IF(IFLGMG.EQ.0) WRITE(IOOUT,917) AMAG,ZONE
917   FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,24X,24HSEARCH ZONE (DEGREE
     *S**2),6X,F7.3)
      IF(IFLGMG.EQ.1) WRITE(IOOUT,918) AMAG,ZONE
918   FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,1X,19H(ARBITRARY DEFAULT),
     *      4X,24HSEARCH ZONE (DEGREES**2),6X,F7.3)
      WRITE (IOOUT,920)
920   FORMAT(1H0,52X,35HREQUIRES (TO DEFEAT DEVICE) AT MOST/
     *             4X,14HFOR NO GREATER,6X,5HUNDER,8X,3HAND,7X,2HAT,4X,
     *35(1H-))
      WRITE(IOOUT,919)
919   FORMAT(8X,5HINPUT,11X,5HINPUT,7X,5HINPUT,5X,5HINPUT,3X,
     *        8HCOMPUTED,2X,11HCONTRAST OR,3X,8HCOMPUTED/6X,
     *        9HDETECTION,6X,11HAMB. ILLUM.,2X,10HSKY/GROUND,2X,
     *        5HRANGE,2X,10HRESOLVABLE,1X,11HTEMP. DIFF.,2X,10HTOTAL PATH
     *        /3X,5HPROB.,2X,9HTIME(SEC),3X,9H(FT CDLS),5X,5HRATIO,5X,
     *        4H(KM),3X,10HCYCLES, RC,2X,9HAT DEVICE,2X,13HTRANSMITTANCE,
     *        13X,8HCOMMENTS/3X,5(1H-),2X,9(1H-),2X,11(1H-),2X,10(1H-),
     *        1X,6(1H-),2X,10(1H-),2X,9(1H-),2X,13(1H-),2X,30(1H-))
      ILINE=23
C ***** BEGIN COMPUTATIONS   FOV,RC,TBAR
11    S=DIM/R
      IF (FOV.LE.0.) FOV=.0001
      TS=1.7*ZONE/FOV**2
      IF(ZONE.GT.9.0.AND.FOV.GT.5.) TS=(1.7*ZONE)/(5.0*FOV)
      IF (TS.LT..5) TS=.5
      CALL CYCLE(PS,PB,AJOB,RC)
      IF (AL.LE.0.) AL=1.E-7
      ALPRNT=AL
      IF (LSC.EQ.13) GOTO 13
      TBAR=0.
      IF (RC.LT.0.1) GOTO 14
      IF((LSC.GT.5.AND.LSC.LT.10).OR.LSC.EQ.11.OR.LSC.EQ.14) GO TO 12
      RCS=RC
      PINF=1.-EXP(-1.7*RCS/6.8)
      IF (PINF.LE.0.) PINF=.0001
      TBAR=0.5*TS*(2.-PINF)/PINF
      GO TO 14
12    TBAR=0.5*TS*(2.0-PS)/PS
      GO TO 14
```

```
C ***** DEVICE #13                                                      ITA03510
13    NAL=1                                                             ITA03520
      ICM=0                                                             ITA03530
      PS=.99                                                            ITA03540
      TBAR=1.8                                                          ITA03550
      C=ACON                                                            ITA03560
      TTOT=1.                                                           ITA03570
      TFNL=TTOT                                                         ITA03580
      GO TO 22                                                          ITA03590
C ***** COMPUTE C BASED ON AMBIENT ILLUMINATION.  INTERPOLATE IF NECESSAITA03600
14    IF (S.LE.0.) S=.0001                                              ITA03610
      IF (RC.LT.0.) RC=0.                                               ITA03620
      RC=RC/S                                                           ITA03630
      IF (AMAG.LE.0.) AMAG=.0001                                        ITA03640
      IF(LSC.EQ.1.OR.LSC.EQ.2) RC=RC/AMAG                               ITA03650
      IF(LSC.EQ.2) AL=0.7*AL                                            ITA03660
C ***** CORRECT FOR FIELD OF VIEW                                       ITA03670
      IF (LSC.LT.6.OR.LSC.EQ.7) GOTO 140                                ITA03680
      IF ((LSC.EQ.6).AND.(MODE.EQ.2)) RC=RC/3.                          ITA03690
      IF ((LSC.EQ.8).AND.(MODE.EQ.1)) RC=RC*3.                          ITA03700
      IF ((LSC.EQ.9).AND.(MODE.EQ.1)) RC=RC*4.                          ITA03710
      IF ((LSC.EQ.11).AND.(MODE.EQ.1)) RC=RC*3.                         ITA03720
      IF ((LSC.EQ.14).AND.(MODE.EQ.1)) RC=RC*3.                         ITA03730
      IF ((LSC.EQ.10).AND.(MODE.EQ.1)) RC=RC*8.                         ITA03740
      IF ((LSC.EQ.12).AND.(MODE.EQ.2)) RC=RC/4.                         ITA03750
140   CONTINUE                                                          ITA03760
      IF(LSC.EQ.2) IAL=0                                                ITA03770
      IF(IAL.GT.0) GO TO 20                                             ITA03780
      AV=100.                                                           ITA03790
      IF(AL.GT.100.) AL=100.                                            ITA03800
      DO 15 I=1,9                                                       ITA03810
      IF(AL.GT.(0.999*AV)) GO TO 16                                     ITA03820
      AV=AV/10.                                                         ITA03830
15    CONTINUE                                                          ITA03840
      I=2                                                               ITA03850
      IF (LSC.GT.5) GOTO 160                                            ITA03860
      NAL=0                                                             ITA03870
      ICM=0                                                             ITA03880
      GO TO 22                                                          ITA03890
C ***** CHECK IF INTERPOLATION NEEDED, IF NOT, GO TO 20                 ITA03900
16    IF (LSC.GT.5) GOTO 160                                            ITA03910
      IF(AL.GT.(1.001*AV)) GO TO 17                                     ITA03920
160   IAL=I                                                             ITA03930
      GO TO 20                                                          ITA03940
C ***** INTERPOLATE                                                     ITA03950
17    NAL1=I-1                                                          ITA03960
      NAL2=I                                                            ITA03970
      ICM1=0                                                            ITA03980
      ICM2=0                                                            ITA03990
      CALL INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA,AMRAL,  ITA04000
     *AMRGM)                                                            ITA04010
      NAL=0                                                             ITA04020
      IF(NAL1.EQ.0.AND.NAL2.EQ.0) GO TO 22                              ITA04030
      NAL=NAL2                                                          ITA04050
      IF(NAL1.GT.0) GO TO 18                                            ITA04040
      ICM=ICM2                                                          ITA04060
      GO TO 21                                                          ITA04070
18    IF(NAL2.GT.0) GO TO 19                                            ITA04080
      NAL=NAL1                                                          ITA04090
      ICM=ICM1                                                          ITA04100
      GO TO 21                                                          ITA04110
19    IF(ICM1.GT.0.AND.ICM2.GT.0) ICM=1                                 ITA04120
      IF(ICM1.LT.0.AND.ICM2.LT.0) ICM=-1                                ITA04130
      GO TO 21                                                          ITA04140
C ***** NO INTERPOLATION NEEDED                                         ITA04150
20    NAL=IAL                                                           ITA04160
      CALL CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM)            ITA04170
21    IF(ICM.EQ.-1) C=.99*C                                             ITA04180
      CALL TREQ(ACON,SOG,C,LSC,TTOT)                                    ITA04190
      IF (ICM.EQ.1) TTOT=1.                                             ITA04200
```

```
      TFNL=TTOT
C ***** PRINT RESULTS
22    IF(NAL.GT.0) GO TO 23
      TFNL=1.
      WRITE(IOOUT,930) PS,TBAR,ALPRNT,SOG,R,TFNL
930   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,4X,6(1H-),7X,5(1H-),5X,
     *       F8.3,6X,28HNOTE- AMBIENT ILLUM. OUTSIDE/91X,
     *       29HDEVICE OPERATIONAL LIMITS. NO/91X,
     *       21HOBSCURATION REQUIRED./1X)
      ILINE=ILINE+3
      GO TO 4
23    IF (TTOT.GE.1..AND.ICM.EQ.0) GOTO 24
      IF(ICM.EQ.0) WRITE(IOOUT,931) PS,TBAR,ALPRNT,SOG,R,RC,C,TTOT
      IF(ICM.EQ.1) WRITE(IOOUT,932) PS,TBAR,ALPRNT,SOG,R,RC,TTOT,C
      IF(ICM.EQ.1) ILINE=ILINE+6
      IF(ICM.EQ.-1) WRITE(IOOUT,933) PS,TBAR,ALPRNT,SOG,R,RC,C,TTOT
      IF(ICM.EQ.-1) ILINE=ILINE+7
      GO TO 4
24    WRITE (IOOUT,934   ) PS,TBAR,ALPRNT,SOG,R,RC,C,TTOT
934   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,
     *       31HNOTE- CONTRAST (OR TEMP. DIFF.)/91X,
     *       31HREQUIRED WOULD EXCEED INTRINSIC/91X,
     *       31HCONTRAST (TEMP. DIFF.). NO OBS-/91X,
     *       17HCURATION REQUIRED/1X)
      ILINE=ILINE+3
      GOTO 4
931   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3)
932   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,7X,5(1H-),5X,F8.3,6X,
     *       29HNOTE- DETECTION PROBABILITY  /91X,
     *       30HREQUIRES CONTRAST (OR TEMP.    /91X,
     *       29HDIFF.) AND RESOLVABLE CYCLES /91X,
     *       27HABOVE LIMIT FOR DEVICE. NO /91X,
     *       26HOBSCURANT REQUIRED. DEVICE/91X,
     *       18HUPPER LIMIT IS C= ,F8.3/1X)
933   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,
     *       30HNOTE- INPUT DETECTION PROBAB- /91X,
     *       30HILITY REQUIRES CONTRAST (OR    /91X,
     *       30HTEMP. DIFF.) BELOW THRESHOLD. /91X,
     *       29HVALUES ASSUMED ARE 99 PERCENT/91X,
     *       29HOF THRESHOLD. ADDITIONAL OBS-/91X,
     *       29HCURANT WILL NOT DECREASE     /91X,
     *       30HDETECTION PROBABILITY.        /1X)
      END
```

```
      SUBROUTINE CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM)
      DIMENSION ALFA(9),BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DATA IPOS,INEG /1,-1/
C  THIS ROUTINE COMPUTES THE REQUIRED CONTRAST AT THE DEVICE FOR A
C     GIVEN NUMBER OF RESOLVABLE CYCLES AT THE DEVICE.  EQUATIONS
C     ARE INVERSIONS OF THOSE OF THE NV&EOL TARGET ACQUISITION
C     (FOR STATIC DETECTION) MODEL.
C
C   INPUTS:  LSC    = DEVICE NUMBER (1-14)
C            RC     = RESOLVABLE CYCLES (DECIMAL)
C            NAL    = AMBIENT ILLUMINATION CATEGORY (1-9)
C
C   OUTPUTS: C      = CONTRAST OR TEMPERATURE DIFFERENCE(DEG K) REQUIRED
C                     BY DEVICE FOR GIVEN RC.
C            ICM    = OPERATIONAL LIMITS FLAG (+1 EXCEEDS LIMIT, -1
C                     BELOW LIMIT, 0 WITHIN RANGE)
C            NAL    = SET TO 0 IF TOO MUCH OR NOT ENOUGH AMBIENT ILLUM.
C
      CFUN(RX,ALF,BET,GAM)=(BET+ALF*RX)/(GAM-RX)
      ICM=0
      IF(NAL.EQ.0) GO TO 9020
C ***** BRANCH TO LSC
5     GO TO (10,10,30,40,50,60,60,80,90,100,1111,1200,9060,1400), LSC
10    GO TO (110,120,130,140,150,160,170,9060,9060), NAL
C ***** DEVICE #1      NAL #1
110   IF(2.74.LT.RC) GO TO 9030
      IF((2.133.LT.RC).AND.(RC.LE.2.74)) GO TO 112
      IF((.26795.LT.RC).AND.(RC.LE.2.133)) GO TO 113
      IF(RC.LE..26795) GO TO 114
      GO TO 9000
112   IEQ=1
      GO TO 9050
113   IEQ=2
      GO TO 9050
114   C=0.015
      GO TO 9015
C ***** DEVICE #1 NAL #2
120   IF(2.74.LT.RC) GO TO 9030
      IF((2.133.LT.RC).AND.(RC.LE.2.74)) GO TO 122
      IF((.0001.LE.RC).AND.(RC.LE.2.133)) GO TO 123
      IF(RC.LT..0001) GO TO 124
      GO TO 9000
122   IEQ=3
      GO TO 9050
123   IEQ=4
      GO TO 9050
124   C=0.025
      GO TO 9015
C ***** DEVICE #1    NAL #3
130   IF(2.29.LT.RC) GO TO 131
      IF((0.49585.LE.RC).AND.(RC.LE.2.29)) GO TO 132
      IF(RC.LT.0.49585) GO TO 133
      GO TO 9000
131   C=0.6324
      GO TO 9035
132   IEQ=5
      GO TO 9050
133   C=0.030
      GO TO 9015
C ***** DEVICE #1      NAL #4
140   IF(1.5219.LT.RC) GO TO 141
      IF((0.313.LE.RC).AND.(RC.LE.1.5219)) GO TO 142
      IF(RC.LT.0.313) GO TO 143
      GO TO 9000
141   C=0.70
      GO TO 9035
142   IEQ=6
      GO TO 9050
143   C=0.05
```

```
      GO TO 9015
C ***** DEVICE #1     NAL #5
150   IF(1.1959.LT.RC) GO TO 9030
      IF((0.0.LE.RC).AND.(RC.LE.1.1959)) GO TO 152
      IF(RC.LT.0.0) GO TO 153
      GO TO 9000
152   IEQ=7
      GO TO 9050
153   C=0.0841
      GO TO 9015
C ***** DEVICE #1       NAL #6
160   IF(0.44767.LT.RC) GO TO 9030
      IF((0.0309.LE.RC).AND.(RC.LE.0.44767)) GO TO 162
      IF(RC.LT.0.0309) GO TO 163
      GO TO 9000
162   IEQ=8
      GO TO 9050
163   C=0.18
      GO TO 9015
C ***** DEVICE #1       NAL #7
170   IF(0.14.LT.RC) GO TO 9030
      IF((0.05.LE.RC).AND.(RC.LE.0.14)) GO TO 172
      IF(RC.LT.0.05) GO TO 173
      GO TO 9000
172   IEQ=9
      GO TO 9050
173   C=0.50
      GO TO 9015
C ***** DEVICE #3
30    GO TO (9060,9060,9060,340,350,360,370,380,390), NAL
C ***** DEVICE #3       NAL #4
340   IF(0.7497.LE.RC.AND.RC.LT.2.941) GO TO 342
      IF (RC.GE.2.941) GOTO 9075
      IF(RC.LT.0.7497) GO TO 9040
      GO TO 9000
342   IEQ=1
      GO TO 9080
C ***** DEVICE #3       NAL #5
350   IF (0.3988.LE.RC.AND.RC.LT.2.4350) GO TO 352
      IF (RC.GE.2.435) GOTO 9075
      IF(RC.LT.0.3988) GO TO 9040
      GO TO 9000
352   IEQ=2
      GO TO 9080
C ***** DEVICE #3       NAL #6
360   IF (0.15965.LE.RC.AND.RC.LT.2.1060) GO TO 362
      IF (RC.GE.2.106) GOTO 9075
      IF(RC.LT.0.15965) GO TO 9040
      GO TO 9000
362   IEQ=3
      GO TO 9080
C ***** DEVICE #3       NAL #7
370   IF (0.05498.LE.RC.AND.RC.LT.2.0375) GO TO 372
      IF (RC.GE.2.0375) GOTO 9075
      IF(RC.LT.0.05498) GO TO 9040
      GO TO 9000
372   IEQ=4
      GO TO 9080
C ***** DEVICE #3       NAL #8
380   IF (0.442.LE.RC.AND.RC.LT.3.2190) GO TO 382
      IF (RC.GE.3.219) GOTO 9075
      IF((RC.LT.0.442).AND.(RC.GT.0.)) GO TO 383
      IF (RC.LE.0.) GOTO 384
      GO TO 9000
382   IEQ=5
      GO TO 9080
383   C=0.57
      GO TO 9015
384   C=0.33
      GOTO 9015
```

```
C ***** DEVICE #3      NAL #9
390    IF (0.172.LE.RC) GO TO 392
       IF(RC.LT.0.172.AND.RC.GT.0.) GO TO 393
       IF (RC.LE.0.) GOTO 394
       GO TO 9000
392    IEQ=6
       GO TO 9080
393    C=0.57
       GO TO 9015
394    C=.33
       GOTO 9015
C ***** DEVICE #4
40     GO TO (9060,9060,9060,440,450,460,470,480,490), NAL
C ***** DEVICE #4      NAL #4
440    IF (1.199.LE.RC.AND.RC.LT.5.2180) GO TO 442
       IF (RC.GE.5.218) GOTO 9075
       IF(RC.LT.1.199) GO TO 9040
       GO TO 9000
442    IEQ=1
       GO TO 9080
C ***** DEVICE #4      NAL #5
450    IF (0.6402.LE.RC.AND.RC.LT.4.2770) GO TO 452
       IF (RC.GE.4.277) GOTO 9075
       IF(RC.LT.0.6402) GO TO 9040
       GO TO 9000
452    IEQ=2
       GO TO 9080
C ***** DEVICE #4      NAL #6
460    IF (0.2449.LE.RC.AND.RC.LT.4.1240) GO TO 462
       IF (RC.GE.4.124) GOTO 9075
       IF(RC.LT.0.2449) GO TO 9040
       GO TO 9000
462    IEQ=3
       GO TO 9080
C ***** DEVICE #4      NAL #7
470    IF (0.08791.LE.RC.AND.RC.LT.3.4200) GO TO 472
       IF (RC.GE.3.420) GOTO 9075
       IF(RC.LT.0.08791) GO TO 9040
       GO TO 9000
472    IEQ=4
       GO TO 9080
C ***** DEVICE #4      NAL #8
480    IF (0.4394.LE.RC.AND.RC.LT.8.5970) GO TO 482
       IF (RC.GE.8.597) GOTO 9075
       IF((RC.LT.0.4394).AND.(RC.GT.0.)) GO TO 483
       IF (RC.LE.0.) GOTO 484
       GO TO 9000
482    IEQ=5
       GO TO 9080
483    C=0.33
       GO TO 9015
484    C=0.07
       GOTO 9015
C ***** DEVICE #1      NAL #9
490    IF (0.1605.LE.RC) GO TO 492
       IF(RC.LT.0.1605.AND.RC.GT.0.) GO TO 493
       IF (RC.LE.0.) GOTO 494
       GO TO 9000
492    IEQ=6
       GO TO 9080
493    C=0.33
       GO TO 9015
494    C=0.07
       GOTO 9015
C ***** DEVICE # 5
50     GO TO (9060,9060,9060,540,550,560,570,580,590), NAL
540    IF (0.9189.LE.RC.AND.RC.LT.2.8060) GO TO 542
       IF (RC.GE.2.806) GOTO 9075
       IF(RC.LT.0.9189) GO TO 9040
       GO TO 9000
```

```
542     IEQ=1
        GO TO 9080
C ***** DEVICE #5       NAL #5
550     IF (0.59035.LE.RC.AND.RC.LT.2.7010) GO TO 552
        IF (RC.GE.2.7010) GOTO 9075
        IF(RC.LT.0.59035) GO TO 9040
        GO TO 9000
552     IEQ=2
        GO TO 9080
C ***** DEVICE #5       NAL #6
560     IF( 0.2881.LE.RC.AND.RC.LT.2.4800) GO TO 562
        IF (RC  .GE.2.480) GOTO 9075
        IF(RC.LT.0.2881) GO TO 9040
        GO TO 9000
562     IEQ=3
        GO TO 9080
C ***** DEVICE #5       NAL #7
570     IF (0.12943.LE.RC.AND.RC.LT.1.6270) GO TO 572
        IF (RC.GE.1.627) GOTO 9075
        IF(RC.LT.0.12943) GO TO 9040
        GO TO 9000
572     IEQ=4
        GO TO 9080
C ***** DEVICE #5       NAL #8
580     IF (0.4949.LE.RC.AND.RC.LT.2.6960) GO TO 582
        IF (RC.GE.2.696) GOTO 9075
        IF(RC.LT.0.4949.AND.RC.GT.0.) GO TO 583
        IF (RC.LE.0.) GOTO 584
        GO TO 9000
582     IEQ=5
        GO TO 9080
583     C=0.33
        GO TO 9015
584     C=0.07
        GOTO 9015
C ***** DEVICE #5       NAL #9
590     IF (0.1894.LE.RC.AND.RC.LT.3.2898) GO TO 592
        IF (RC.GE.3.2898) GOTO 9075
        IF(RC.LT.0.1894.AND.RC.GT.0.) GO TO 593
        IF (RC.LE.0.) GOTO 594
        GO TO 9000
592     IEQ=6
        GO TO 9080
593     C=0.33
        GO TO 9015
594     C=0.07
        GOTO 9015
C ***** DEVICES #6 AND #7
60      IF(1.7934.LT.RC) GO TO 601
        IF((0.92376.LT.RC).AND.(RC.LE.1.7934)) GO TO 602
        IF((0.11022.LE.RC).AND.(RC.LE.0.92376)) GO TO 603
        IF(RC.LT.0.11022) GO TO 9070
        GO TO 9000
601     C=8.38
        GO TO 9035
602     C=(.172363*RC-0.0392775)/(1.82560-RC)
        GO TO 9020
603     C=(0.09772*RC)/(1.-0.34779*RC)
        GO TO 9020
C ***** DEVICE #8
80      IF(4.702.LT.RC) GO TO 801
        IF((0.996.LE.RC).AND.(RC.LE.4.702)) GO TO 802
        IF(RC.LT.0.996) GO TO 803
        GO TO 9000
801     C=2.18
        GO TO 9035
802     C=(0.0298*RC)/(1.-0.199*RC)
        GO TO 9020
803     C=0.037
        GO TO 9015
```

```
C ***** DEVICE #9
90     IF(0.37181.LE.RC.AND.RC.LT.9.14027) GO TO 901
       IF (RC.GE.9.14027) GOTO 9075
       IF(RC.LT.0.37181) GO TO 9070
       GO TO 9000
901    C=(0.0289*RC)/(1.-0.1092*RC)
       GO TO 9020
C ***** DEVICE #10
100    IF(NAL.GT.2) GO TO 9060
       IF(4.227.LE.RC.AND.RC.LT.91.05956) GO TO 1010
       IF (RC.GE.91.05956) GOTO 9075
       C=0.01
C   NEXT STATEMENT  IS LIMINAL CONTRAST ADJUSTMENT
       C=C/1.63
       GO TO 9015
1010   C=(0.0303*RC)/(13.43-0.1473*RC)
C   NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
       C=C/1.63
       GO TO 9020
C ***** DEVICE # 11
1111   IF(0.5057.LE.RC.AND.RC.LT.7.73353) GO TO 1110
       IF (RC.GE.7.73353) GOTO 9075
       GO TO 9070
1110   C=(0.0207*RC)/(1.-0.1291*RC)
       GO TO 9020
C ***** DEVICE # 12
1200   IF(NAL.GT.2) GO TO 9060
       IF(3.48.LE.RC.AND.RC.LT.31.48113) GO TO 1210
       IF (RC.GE.31.41883) GOTO 9075
       C=0.01
C   NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
       C=C/1.63
       GO TO 9015
1210   C=(0.0206*RC)/(8.06-0.2559*RC)
C   NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
       C=C/1.63
       GO TO 9020
C ***** DEVICE # 14
1400   IF(0.9098.LE.RC.AND.RC.LT.4.57711) GO TO 1410
       IF (RC.GE.4.57711) GOTO 9075
       C=0.037
       GO TO 9015
1410   C=(0.0297567*RC)/(0.91287-0.1991449*RC)
       GO TO 9020
9000   WRITE(IOOUT,9010) LSC,NAL,RC
9010   FORMAT(5X,46HBAD PARAMETER PASSED TO SUBRTN CINV **** LSC= ,I3,
     *          6H NAL= ,I3,5H RC= ,F10.4)
9015   ICM=INEG
9020   RETURN
9030   C=0.80
9035   ICM=IPOS
       GO TO 9020
9040   C=0.02
       GO TO 9015
9050   C=CFUN(RC,ALFA(IEQ),BETA(IEQ),GAMA(IEQ))
       GO TO 9020
9060   NAL=0
       GO TO 9020
9070   C=0.0112
       GO TO 9015
9075   C=100.
       GOTO 9035
9080   L=LSC-2
       C=CFUN(RC,AMRAL(IEQ,L),0.,AMRGM(IEQ,L))
       GOTO 9020
       END
```

```
      SUBROUTINE INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA,
     *AMRAL,AMRGM)
C
C  INTERPOLATION ON THE SURFACE IN C, RC, AMBIENT ILLUMINATION SPACE.
C     THE USER INPUTS A VALUE OF RC, AND AN AMBIENT ILLUMINATION THAT
C     LIES BETWEEN ONE PAIR OF MODELED RC VS C CURVES AT TWO DISCRETELY
C     MODELED AMBIENT ILLUMINATIONS. THE TECHNIQUE IS COMPLICATED BY A
C     REQUIREMENT TO REPRODUCE ALMOST EXACTLY THE INTERPOLATED VALUE
C     FROM RC VS C OF THE NVL TARGET ACQUISITION MODEL.  THUS, WHILE
C     RC IS INPUT IN THIS INVERSION MODEL, THE INTERPOLATION IS BETWEEN
C     RC VALUES AT CONSTANT C OVER TWO AMB. ILLUM. REGIONS. THE
C     THRESHOLDS OR OPERATIONAL LIMITS OF THE DEVICE ARE TREATED SEPAR-
C     ATELY.  IN ALL, FOUR TYPES OF INTERPOLATION SITUATIONS ARE
C     UTILIZED.
C
      DIMENSION IRGNS(21),INDEX(21),RCL(25),RCU(25),CU(25),CL(25),
     *IEQ1(36),IEQ2(36),RLW(36),RUP(36),CLW(36),CUP(36),ALFA(9),
     *BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C** NUMBER OF C REGIONS FOR INTERPOLATION CONSIDERATION, DEV. 1-5
      DATA IRGNS /4,4,3,3,2,2,1,1,1,2,1,1,1,1,2,1,1,1,1,2,1/
C** STARTING INDEX FOR EACH DEVICE, AMB.ILLUM. PAIR.
      DATA INDEX /1,5,9,12,15,17,19,20,21,22,24,25,26,27,28,30,31,
     *32,33,34,36/
C** LOWER LIGHT LEVEL THRESHOLDS OF RC.
      DATA RCL  /
     *.0001 ,.26795,.49585,.313  ,0.    ,.0309 ,.05   ,.7497 ,.3988 ,
     *.15965,.055  ,.442  ,.1742 ,1.199 ,.6402 ,.2449 ,.0879 ,.4394 ,
     *.1605 ,.9189 .5903 ,.288   ,.1294 ,.4949 ,.1894 /
C** UPPER LIGHT LEVEL THRESHOLDS OF RC
      DATA RCU  /
     *2.74  ,2.74  ,2.29  ,1.5219,1.1959,.44767,.14   ,2.941 ,2.435 ,
     *2.106 ,2.0375,3.219 ,3.219 ,5.218 ,4.277 ,4.124 ,3.42  ,8.579 ,
     *8.597 ,2.806 ,2.701 ,2.48  ,1.627 ,2.696 ,3.2898/
C** VALUE OF C AT LOWER THRESHOLDS.
      DATA CL   /
     *.015 ,.025 ,.03  ,.05  ,.0841,.18  ,.50  ,.02  ,.02  ,.02  ,
     *.02  ,.57  ,.57  ,.02  ,.02  ,.02  ,.02  ,.33  ,.33  ,.02  ,
     *.02  ,.02  ,.02  ,.33  ,.33  /
C** VALUE OF C AT UPPER THRESHOLD
      DATA CU   /
     *.80  ,.80  ,.6324,.70  ,.80  ,.80  ,.80  ,100. ,100. ,100. ,
     *100. ,100. ,16.88,100. ,100. ,100. ,100. ,100. ,16.17,100. ,
     *100. ,100. ,100. ,100. ,100. /
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR UPPER AMB. ILL.
      DATA IEQ1 /2,2,1,1,4,4,3,3,5,5,0,6,6,0,7,7,8,8,1,2,3,4,4,5,1,2,3,
     *           4,4,5,1,2,3,4,4,5/
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR LOWER AMB ILL.
      DATA IEQ2 /-1,4,4,3,-1,5,5, 0,-1,6,6,-1,7,7,-1,8,-1,9,2,3,4,-1,5,
     *           6,2,3,4,-1,5,6,2,3,4,-1,5,6/
C** LOWEST RC VALUE IN EACH REGION.
      DATA RLW  /
     *0.    ,.26794,2.118 ,2.133 ,.26794,.4958 ,1.8439,2.29  ,.312  ,
     *.312  ,1.5119,0.    ,0.    ,1.1758,0.    ,.03089,.03089,.05   ,
     *.3987 ,.15964,.054  ,.054  ,.442  ,.174  ,.64   ,.244  ,.087  ,
     *.087  ,.4394 ,.1605 ,.59   ,.287  ,.129  ,.129  ,.4948 ,.1894 /
C** LARGEST RC VALUE IN EACH REGION.
      DATA RUP  /
     *.67890,2.1335,2.1637,2.74  ,.61280,2.133 ,2.5764,2.74  ,.739  ,
     *2.29  ,2.29  ,.84083,1.5219,1.5219,.712  ,1.1959,.3751 ,.4477 ,
     *2.942 ,2.435 ,2.106 ,.9023 ,3.219 ,3.219 ,5.218 ,4.277 ,4.124 ,
     *1.0396,8.597 ,8.597 ,2.81  ,2.71  ,2.49  ,.9575 ,2.696  ,3.29 /
C** LOWEST VALUE OF C OVER EACH REGION.
      DATA CLW /
     *.015  ,.025  ,.250  ,.270  ,.025  ,.030  ,.270  ,.6324 ,.030  ,
     *.050  ,.6324 ,.050  ,.084107,.70  ,.0841 ,.180  ,.180  ,.500  ,
     *.020  ,.020  ,.020  ,.020  ,.570  ,.570  ,.020  ,.020  ,.020  ,
     *.020  ,.330  ,.330  ,.020  ,.020  ,.020  ,.020  ,.330  ,.330  /
C** LARGEST VALUE OF C OVER EACH REGION.
      DATA CUP  /
```

```
     *.025  ,.250   ,.270  ,.800   ,.030   ,.270   ,.6324 ,.800   ,.050   ,
     *.6324 ,.700   ,.084107,.70   ,.800   ,.180   ,.800  ,.500   ,.800   ,
     *100.  ,100.   ,100.  ,.570   ,100.   ,16.88  ,100.  ,100.   ,100.   ,
     *.330  ,100.   ,16.17 ,100.   ,100.   ,100.   ,.330  ,100.   ,100.   /
C** SET LIGHT LEVELS, FRACTIONAL INTERPOLANT.
      ICM1=0
      ICM2=0
      IF (NAL1.EQ.NAL2) GOTO 10
      IF (LSC.GT.5) GOTO 10
      AL1=10.**(3-NAL1)
      AL2=10.**(3-NAL2)
      FACTR=(AL-AL2)/(AL1-AL2)
      DFAC=1.-FACTR
      GOTO (100,100,200,200,200),LSC
10    CALL CINV(LSC,RC,NAL1,C,ICM1,ALFA,BETA,GAMA,AMRAL,AMRGM)
20    NAL2=0
      RETURN
30    CALL CINV(LSC,RC,NAL2,C,ICM2,ALFA,BETA,GAMA,AMRAL,AMRGM)
40    NAL1=0
      RETURN
C** DEVICES 1,2
100   IF (NAL2.LE.7) GOTO 105
      IF (NAL1.EQ.7.AND.FACTR.GT.0.75) GOTO 10
      NAL1=0
      GOTO 20
105   NRG=IRGNS(NAL1)
      IDX=INDEX(NAL1)
      LIM1=NAL1
      LIM2=NAL2
      GOTO 500
C** DEVICES 3,4,5
200   IF (NAL1.GE.4) GOTO 205
      IF (NAL2.EQ.4.AND.FACTR.LT.0.3) GOTO 30
      NAL2=0
      GOTO 40
205   IS=6+(LSC-3)*5+(NAL1-3)
      NRG=IRGNS(IS)
      IDX=INDEX(IS)
      LIM1=7+(LSC-3)*6+(NAL1-3)
      LIM2=LIM1+1
      BVAL=0.
      IDV=LSC-2
C** FIRST, CHECK LIMIT.
500   IF (RC.GE.RCL(LIM1).AND.RC.GE.RCL(LIM2)) GOTO 510
      CALL CASE3(FACTR,CNOT,RCN,RCL(LIM1),RCL(LIM2),CL(LIM1),CL(LIM2))
      IF (RC.GT.RCN) GOTO 520
      CV=CNOT
      ICM1=-1
      ICM2=-1
      GOTO 800
510   IF (RC.LE.RCU(LIM1).AND.RC.LE.RCU(LIM2)) GOTO 520
      CALL CASE3(FACTR,CNOT,RCN,RCU(LIM1),RCU(LIM2),CU(LIM1),CU(LIM2))
      IF (RC.LT.RCN) GOTO 520
      CV=CNOT
      ICM1=1
      ICM2=1
      GOTO 800
C** RC IS IN BOUNDS OF OPERATIONAL LIMITS, BUT NOT NECESSARILLY
C   BETWEEN CURVES.  TEST SUB-REGIONS IN TURN.
520   DO 600 I=1,NRG
      J=IDX+I-1
      IF (RC.LT.RLW(J).OR.RC.GT.RUP(J)) GOTO 600
C** RC IS IN SUB-REGION. NOW TEST C
      CV=0.
      CR1=0.
      CR2=0.
      IQ1=IEQ1(J)
      IQ2=IEQ2(J)
      IF (IQ1.GT.0) GOTO 530
      RCN=RCU(LIM1)
```

```
      CNOT=CU(LIM1)
      IF (IQ1.LT.0) RCN=RCL(LIM1)
      IF (IQ1.LT.0) CNOT=CL(LIM1)
      IF (LSC.LE.2) CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ2),
     *BETA(IQ2),GAMA(IQ2))
      IF (LSC.GT.2)CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,
     *AMRAL(IQ2,IDV),BVAL,AMRGM(IQ2,IDV))
      IF (IFLG.NE.0) GOTO 600
      IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600
      IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600
      IF (RCN.EQ.RCU(LIM1)) ICM1=1
      IF (RCN.EQ.RCL(LIM1)) ICM1=-1
      GOTO 800
530   IF (IQ2.GT.0) GOTO 540
      RCN=RCU(LIM2)
      CNOT=CU(LIM2)
      IF (IQ2.LT.0) RCN=RCL(LIM2)
      IF (IQ2.LT.0) CNOT=CL(LIM2)
      IF (LSC.LE.2) CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ1),
     *BETA(IQ1),GAMA(IQ1))
      IF (LSC.GT.2)CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,
     *AMRAL(IQ1,IDV),BVAL,AMRGM(IQ1,IDV))
      IF (IFLG.NE.0) GOTO 600
      IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600
      IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600
      IF (RCN.EQ.RCU(LIM2)) ICM2=1
      IF (RCN.EQ.RCL(LIM2)) ICM2=-1
      GOTO 800
540   IF(LSC.LE.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,ALFA(IQ1),ALFA(IQ2),
     *BETA(IQ1),BETA(IQ2),GAMA(IQ1),GAMA(IQ2))
      IF(LSC.GT.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,AMRAL(IQ1,IDV),
     *AMRAL(IQ2,IDV),BVAL,BVAL,AMRGM(IQ1,IDV),AMRGM(IQ2,IDV))
      IF (IFLG.GE.2) GOTO 600
      CV=CR1
      IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800
      IF (IFLG.EQ.1) GOTO 600
      CV=CR2
      IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800
600   CONTINUE
C** NO VALUES MET INTERPOLATION CRITERIA.
C** FINAL TEST IS IN THE LIMITING REGIONS
640   CONTINUE
      WRITE (IOOUT,1000) LSC,AL1,AL2
1000  FORMAT(1X,40H*** IN ITAM INTAL ROUTINE, DEVICE LSC = ,I2,
     */1X,48H*** COULD NOT INTERPOLATE BETWEEN AMBIENT ILLUM ,F11.6,
     *5H AND ,F11.6/1X,39H*** UPPER AMBIENT ILLUM. VALUE ASSUMED.)
      GOTO 10
800   C=CV
      RETURN
      END
```

```
      SUBROUTINE CASE1(FACTR,RC,CR1,CR2,IFLG,ALP1,ALP2,BET1,BET2,
     *GAM1,GAM2)
C
C FINDS CONTRAST C INTERPOLATING ON RC VS C.  RETURNS AT MOST TWO
C  ROOTS CR1, CR2. IFLG=1 IF ROOTS ARE IDENTICAL. IFLG=2 IF ROOTS
C  ARE COMPLEX.
C
      DFAC=1.-FACTR
      A1=RC-GAM1*FACTR-GAM2*DFAC
      A2=RC*(ALP1+ALP2)+(BET1-ALP2*GAM1)*FACTR+(BET2-ALP1*GAM2)*DFAC
      A3=ALP1*ALP2*RC+ALP2*BET1*FACTR+ALP1*BET2*DFAC
      IFLG=2
      IF (A1.LT.-1.E-10.OR.A1.GT.1.E-10) GOTO 10
      IF (A2.EQ.0.) RETURN
      IFLG=1
      CR1=-A3/A2
      CR2=CR1
      RETURN
 10   DISCR=A2*A2-4.*A1*A3
      IF (DISCR.LT.0.) RETURN
      IF (DISCR.EQ.0.) IFLG=1
      IF (DISCR.GT.0.) IFLG=0
      VA=SQRT(DISCR)/(2.*A1)
      VB=-A2/(2.*A1)
      SG=-1.
      IF (A1.LT.0.) SG=1.
      CR1=VB+SG*VA
      CR2=VB-SG*VA
      RETURN
      END
```

```
      SUBROUTINE CASE2(FACTR,RC,C,IFLG,CP,RNOT,CNOT,ALP,BET,GAM)
C
C FINDS INTERPOLATED CONTRAST C FOR CASE OF SINGLE FIXED RC AT
C SOME LIMIT.
C
      IFLG=1
      IF (FACTR.NE.1.) GOTO 10
      IF (RC.NE.RNOT) RETURN
      GOTO 90
10    IF (RC.EQ.RNOT) GOTO 25
      DIFF=RC-RNOT*FACTR
      IF (DIFF.NE.0.) GOTO 20
      IF (GAM.EQ.0.) RETURN
      CP=BET/GAM
      GOTO 50
20    VA=BET*(1.-FACTR)+ALP*DIFF
      VB=GAM*(1.-FACTR)-DIFF
      IF (VB.EQ.0.) GOTO 90
      CP=VA/VB
      GOTO 50
25    IF (GAM.EQ.RC) RETURN
      CP=(ALP*RC+BET)/(GAM-RC)
      GOTO 60
50    RP=(GAM*CP-BET)/(ALP+CP)
      C=CNOT+(CP-CNOT)*(RC-RNOT)/(RP-RNOT)
      GOTO 100
60    C=CNOT*FACTR+CP*(1.-FACTR)
      GOTO 100
90    C=CNOT
      CP=CNOT
100   IFLG=0
      RETURN
      END
```

```
      SUBROUTINE CASE3(FACTR,C,RCN,RNOT1,RNOT2,CNOT1,CNOT2)
C
C FINDS INTERPOLATED C FOR CASE OF BOTH RC VALUES FIXED AT
C LIMITS.
C
      RCN=RNOT1*FACTR+RNOT2*(1.-FACTR)
      IF (RNOT1.EQ.RNOT2) GOTO 10
      C=CNOT1+(CNOT2-CNOT1)*(RCN-RNOT1)/(RNOT2-RNOT1)
      RETURN
10    C=CNOT1*FACTR+CNOT2*(1.-FACTR)
      RETURN
      END
```

```
      SUBROUTINE CYCLE(PS,PB,AJOB,RC)
C
C   THIS ROUTINE COMPUTES THE NORMALIZED NUMBER OF RESOLVABLE CYCLES
C      FOR GIVEN CUMULATIVE NORMAL DISTRIBUTION OF PROBABILITY OF
C      DETECTION.
C
C   INPUTS:   PS   = CUMULATIVE PROBABILITY OF DETECTION.
C             PB   = STANDARD DEVIATION FOR PS (.632 )
C             AJOB = MEAN RESOLVABLE CYCLES FOR 50 PERCENT PROBABILITY
C                    OF DETECTION.
C   OUTPUTS: RC    = RESOLVABLE CYCLES REQUIRED FOR PS
C   SUBROUTINE: CYCLE CALLED BY ITAM.  CALLS NONE.
C
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DATA A0,A1,A2 /2.515517,0.802853,0.010328/
      DATA B1,B2,B3 /1.432788,0.189269,0.001308/
      ICASE=0
      IF((0.0.LE.PS).AND.(PS.LE.0.0000003)) GO TO 10
      IF((.9999997.LE.PS).AND.(PS.LE.1.)) GO TO 20
      IF((.0000003.LT.PS).AND.(PS.LE.0.5)) GO TO 30
      IF((.5.LT.PS).AND.(PS.LT.0.9999997)) GO TO 40
C ***** ERROR ROUTINE
      WRITE(IOOUT,900) PS
      RETURN
C ***** SPECIAL CASE I
10    X=-5.
      IF (PS.EQ.0.) X=-45.
      GO TO 60
C ***** SPECIAL CASE II
20    X=5.
      IF (PS.EQ.1.) X=45.
      GO TO 60
C ***** CASE I
30    XX=SQRT(ALOG(1/(PS*PS)))
      ICASE=1
      GO TO 50
C ***** CASE II
40    XX=SQRT(ALOG(1./(1.-PS)**2))
50    X=XX-(A0+A1*XX+A2*(XX*XX))/(1.+B1*XX+B2*(XX*XX)+B3*(XX*XX*XX))
      IF(ICASE.EQ.1) X=X*(-1.)
60    RC=AJOB+AJOB*PB*X
      RETURN
900   FORMAT(2X,39HERROR - INCORRECT PS VALUE PASSED  PS =  ,F10.4)
      END
```

```
      SUBROUTINE TREQ(ACON,SOG,C,LSC,TRQ)
C
C  THIS ROUTINE COMPUTES THE TRANSMISSION REQUIRED TO MATCH GIVEN
C     TARGET AND DETECTOR CONTRASTS, WITH MODIFICATION BY SKY/
C     GROUND RATIO FOR NON-THERMAL DEVICES.
C
C  INPUTS:  ACON  = INTRINSIC CONTRAST OF TARGET, OR TEMP. DIFF.
C                   (DEG K) FOR THERMAL.
C           SOG   = SKY/GROUND RATIO.
C           C     = CONTRAST OR TEMP. DIFF. SEEN AT DEVICE
C           LSC   = DEVICE NUMBER.
C  OUTPUTS: TRQ   = REQUIRED TRANSMISSION TO REDUCE ACON TO C.
C  SUBROUTINE: TREQ CALLED BY ITAM.  CALLS NONE.
C
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      IF((LSC.LT.1).OR.(LSC.GT.14)) GO TO 30
      IF(LSC.GT.5) GO TO 20
C ***** NON-THERMAL
10    IF (SOG.EQ.0. .OR. C.EQ.0.) GOTO 45
      TRQ=1./(1.+((ACON-C)/(C*SOG)))
      GOTO 50
20    IF((LSC.EQ.10).OR.(LSC.EQ.12)) GO TO 10
      IF(LSC.EQ.13) GO TO 30
C ***** THERMAL
      IF (ACON.EQ.0.) GOTO 25
      TRQ=C/ACON
      GOTO 50
25    TRQ=1.
      GOTO 50
C ***** ERROR ROUTINE
30    WRITE(IOOUT,100) LSC
      TRQ=1.
100   FORMAT(5X,19HINPUT ERROR  LSC = ,I2)
      RETURN
45    TRQ=0.
50    IF (TRQ.LT.0.) TRQ=0.
      IF (TRQ.GT.1.) TRQ=1.
      RETURN
      END
```

```
      SUBROUTINE FCLOUD(LAMBDA,TRANS,IERR)
      REAL LAMBDA,KAPPA,LB0,LP
      REAL L,M,N,L0,M0,N0,L0S,M0S,N0S,L0M,L0N,N0M,MXLY,NXLZ,MZNY
      DIMENSION IALPH(10)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /FGEOM/XC,YC,ZC,AE,BE,CE,XR,YR,ZR,XS,YS,ZS
      COMMON /OPT/ INDEXP,ETA,KAPPA,W0,THETA0,PHI0,LD,TAUBAR,
     +RHO,LB0,TMPA,TMPC
      COMMON /GEOMET/PTS(15),IGEOSW
      DATA IZERO/0/
      DATA IALPH/2HCP  ,2HRP  ,2HSP  ,2HAX  ,2HCL  ,2HAT  ,2HBK  ,
     +2HSA  ,2HLU  ,2HGO  /
      ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)
C***********************************************************************
C
C   SUBROUTINE FCLOUD COMPUTES THE FOLLOWING QUANTITIES FOR A FINITE
C   ELLIPSOIDAL CLOUD:
C
C      S = LENGTH OF OPTICAL PATH IN CLOUD
C      TRANS = BEAM TRANSMITTANCE THROUGH CLOUD
C      LP = PATH RADIANCE
C      TC = CONTRAST TRANSMITTANCE.
C
C   INPUT OF COMPUTATIONAL PARAMETERS TAKES PLACE THROUGH AN ORDER
C   INDEPENDENT READ OF A GROUP OF RECORDS IDENTIFIED BY THE FOLLOWING
C   MNEMONICS (EACH RECORD IS FORMAT (A4,1X,5(E10.4,1X)) ) :
C
C   MNEUMONIC       VARIABLES READ            DESCRIPTION
C   ---------       --------------            -----------
C
C     CPOS            XC,YC,ZC             CLOUD CENTER POSITION
C     RPOS            XR,YR,ZR             RECEIVER POSITION
C     SPOS            XS,YS,ZS             SOURCE POSITION
C     AXES            AE,BE,CE             SEMI-AXES OF CLOUD ELLIPSOID
C
C     CLDS    INDEXP,ETA,KAPPA,W0,TMPC     CLOUD AEROSOL PARAMETERS
C     ATMO            TAUBAR,TMPA          ATMOSPHERIC PARAMETERS
C     BKGR            RHO,LB0              BACKGROUND PARAMETERS
C     SANG            THETA0,PHI0          SOLAR ANGLES
C     LUND            LD                   LUNAR DAY
C
C     GO                                   TERMINATES READ
C
C   ** NOTE : THE GO SENTINEL CARD MUST BE THE LAST CARD READ
C
C   THIS INPUT DATA IS STORED FOR LATER USE IN COMMON
C   BLOCKS /FGEOM/ AND /OPT/.
C
C   /FGEOM/ INPUT PARAMETERS:
C
C      (XC,YC,ZC) = CENTER OF ELLIPSOID
C      (AE,BE,CE) = SEMI-AXES OF ELLIPSOID
C      (XR,YR,ZR) = COORDINATES OF RECEIVER LOCATION
C      (XS,YS,ZS) = COORDINATES OF SOURCE LOCATION.
C
C   /OPT/ INPUT PARAMETERS:
C
C      INDEXP = PHASE FUNCTION IDENTIFIER
C                         =0,  USER SUPPLIED
C                         =1,  MARITIME ARCTIC, VIS=0.1 TO 2.0 KM
C                         =2,  MARITIME POLAR, VIS=0.2 KM
C                         =3,  MARITIME POLAR, VIS=02. KM
C                         =4,  CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM
C                         =5,  WHITE PHOSPHORUS
C                         =6,  HEXACHLOROETHANE
C                         =7,  FOG OIL
C                         =8,  DUST (MODERATE AEROSOL LOADING)
C                         =9,  DUST (HEAVY AEROSOL LOADING)
C                         =10, MARITIME MODEL B, VIS=5KM, RH=95%
C                         =11, MARITIME MODEL B, VIS=10KM,RH=90%
```

```
C                  =12, MARITIME MODEL B, VIS=50KM,RH=50%
C      ETA = FORWARD SCATTERING PARAMETER
C            ETA MAY ALSO BE CALCULATED BY DEFAULT (I.E. INPUT ETA =0.0).
C            IN THIS CASE INDEXP SHOULD BE THE NEGATIVE OF THE PHASE
C            FUNCTION FOR WHICH ETA IS DESIRED.
C      KAPPA = VOLUME EXTINCTION COEFFICIENT (KM-1)
C      W0 = SINGLE SCATTERING ALBEDO
C      (THETA0,PHI0) = SOLAR ANGLES (DEGREES)
C      LD = LUNAR DAY (INTEGER BETWEEN 0 AND 28)
C      TAUBAR = ATMOSPHERIC OPTICAL THICKNESS ABOVE CLOUD
C      RHO = BACKGROUND SURFACE ALBEDO
C      LB0 = BACKGROUND RADIANCE
C      TMPA = TEMPERATURE OF ATMOSPHERE (DEG.C)
C      TMPC = TEMPERATURE OF CLOUD (DEG.C).
C
C   ALL LENGTH UNITS ARE KM.  PROGRAM FLOW IS CONTROLLED BY THE
C   VARIABLES TMPA, INDEXP, AND, IMPLICITLY, BY ISW .
C   IF TMPA >= -99.0, THIS SPECIFIES
C   A THERMAL COMPUTATION, WHICH IS PERFORMED IN SUBROUTINE THRMCL:
C   IF TMPA < -99.0 THEN A SCATTERING COMPUTATION IS PERFORMED.
C   IF INDEXP < 0,  THIS IS A MULTIPLE SCATTERING COMPUTATION, WHICH
C   IS DONE IN SUBROUTINE MSCLD; IF ETA HAS BEEN INPUT AS ZERO, THEN
C      INDEXP SHOULD BE THE NEGATIVE OF THE PHASE FUNCTION INDENTIFIER,
C      SO THAT ETA WILL BE FOUND FROM THE PROPER PHASE FUNCTION.
C      IF INDEXP > 0 A SINGLE SCATTERING
C   COMPUTATION IS CARRIED OUT IN SUBROUTINE SSCLD.  IN THIS CASE
C   INDEXP ALSO SPECIFIES THE PHASE FUNCTION TO BE USED, WITH
C   INDEXP = I SELECTING THE I-TH PHASE FUNCTION.  THE VALUE OF
C   ISW OCCURRING IN THE SUBROUTINE PARAMETER LIST INDICATES WHETHER
C   CERTAIN PARAMETERS ARE THE SAME AS IN THE PREVIOUS CALL TO FCLOUD,
C   AS FOLLOWS.
C
C   ISW IS SET TO 0 WHEN THE FOLLOWING CONDITIONS ARE ENCOUNTERED :
C        A) ALL 9 DATA CARDS ARE READ
C        B) ANY ONE OF CARDS 1-4 AND ANY ONE OF CARDS 5-9
C           (AS LISTED IN THE ORDER ABOVE) ARE READ
C   ISW IS SET TO 2 IF NONE OF CARDS 1-4 IS READ.
C   ISW IS SET TO 1 IF NONE OF CARDS 5-9 IS READ.
C   ISW DEFAULTS TO 2 IF NOTHING IS READ (OTHER THAN THE GO CARD).
C
C       ISW = 2 => PARAMETERS IN COMMON /FGEOM/ ARE THE SAME AS IN
C                  PREVIOUS CALL: SKIP PRELIMINARY GEOMETRICAL
C                  CALCULATIONS
C       ISW = 1 => PARAMETERS IN COMMON /OPT/ ARE THE SAME AS IN
C                  PREVIOUS CALL: SKIP COMPUTATIONS INVOLVING
C                  ONLY THESE PARAMETERS
C       ISW = 0 => NEW PARAMETERS IN BOTH /FGEOM/ AND /OPT/:
C                  NO CALCULATIONS ARE SKIPPED IN THIS CASE.
C
C   SUBROUTINES CALLED FROM FCLOUD:
C
C      THRMCL          FOR THERMAL CALCULATION OF PATH RADIANCE
C      MSCLD           FOR MULTIPLE SCATTERING CALCULATION OF
C                      PATH RADIANCE
C      SSCLD           FOR SINGLE SCATTERING CALCULATION OF PATH
C                      RADIANCE
C      ILLUM           COMPUTES THE EXTRATERRESTRIAL SOLAR OR
C                      LUNAR IRRADIANCE.  NOT NEEDED FOR THERMAL
C                      CALCULATIONS
C      PFN             RETURNS A PHASE FUNCTION VALUE FOR USE BY
C                      SSCLD.
C
C   GROUND RADIANCE CAN BE SPECIFIED IN ONE OF TWO WAYS:
C      1.)  BY GIVING LB0 A NONZERO VALUE, OR
C      2.)  AS REFLECTED SOLAR IRRADIANCE, USING THE VALUE OF
C           RHO INPUT TO THE PROGRAM.
C
C   THE VALUE OF LD SPECIFIES WHETHER IT IS DAY OR NIGHT, I. E.
C   WHETHER SOLAR OR LUNAR EXTRATERRESTRIAL IRRADIANCE IS TO BE
C   USED.  IF LD = 0 THEN SOLAR IRRADIANCE IS CALCULATED, AND IF
```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF

UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

4 - 6
AD
A114417

```
C   1 < LD < 28 THE IRRADIANCE VALUE IS FOR LUNAR DAY LD.
C
C***********************************************************************
C
C   INITIALIZATION OF INPUT DATA.
C
      IF (IZERO.NE.0) WRITE (IOOUT,99)
99    FORMAT(1H1)
      IF(IZERO.NE.0) GO TO 477
      XC=0.0
      YC=0.0
      ZC=0.0
      XR=0.0
      YR=0.0
      ZR=0.0
      XS=0.0
      YS=0.0
      ZS=0.0
      AE=0.0
      BE=0.0
      CE=0.0
      INDEXP=0
      ETA=0.0
      KAPPA=0.0
      W0=0.0
      TMPC=0.0
      TAUBAR=0.0
      TMPA=0.0
      RHO=0.0
      LB0=0.0
      THETA0=0.0
      PHI0=0.0
      LD=0
      ISW=0
      IZERO=1
  477 CONTINUE
      IFLG=2
      IFLO=1
      DO 360 K=1,10
      READ(IOIN,334)IA,IA2,R1,R2,R3,R4,R5
  334 FORMAT(2A2,1X,5(E10.4,1X))
      DO 333 I=1,11
      IF(IA.NE.IALPH(I)) GO TO 333
      IND=I
      IF(IND.EQ.10) GO TO 361
  333 CONTINUE
      IF(K.EQ.10.AND.IND.NE.10) GO TO 358
      IF(IND.LT.5)IFLG=0
      IF(IND.GE.5.AND.IND.LE.9)IFLO=0
      IF(IND.EQ.11) GO TO 355
      IF(IND.LT.5) GO TO (341,342,343,344),IND
      INDM4=IND-4
      GO TO (345,346,347,348,349),INDM4
  341 XC=R1
      YC=R2
      ZC=R3
      GO TO 360
  342 XR=R1
      YR=R2
      ZR=R3
      GO TO 360
  343 XS=R1
      YS=R2
      ZS=R3
      GO TO 360
  344 AE=R1
      BE=R2
      CE=R3
      GO TO 360
  345 INDEXP=IFIX(R1)
```

```
      IF (R2.GT..001) ETA=R2                                            FCL02020
      KAPPA=R3                                                          FCL02030
      W0=R4                                                             FCL02040
      TMPC=R5                                                           FCL02050
      GO TO 360                                                         FCL02060
  346 TAUBAR=R1                                                         FCL02070
      TMPA=R2                                                           FCL02080
      GO TO 360                                                         FCL02090
  347 RHO=R1                                                            FCL02100
      LB0=R2                                                            FCL02110
      GO TO 360                                                         FCL02120
  348 THETA0=R1                                                         FCL02130
      PHI0=R2                                                           FCL02140
      GO TO 360                                                         FCL02150
  349 LD=IFIX(R1)                                                       FCL02160
      GO TO 360                                                         FCL02170
  355 WRITE(IOOUT,357)                                                  FCL02180
  357 FORMAT(1H0,20X,46H***FCLOUD ERROR*** INPUT RECORD DETECTED WHICH, FCL02190
     + 44HDOES NOT CORRESPOND TO CORRECT INPUT FORMAT,/)                FCL02200
      GO TO 360                                                         FCL02210
  358 WRITE(IOOUT,359)                                                  FCL02220
  359 FORMAT(1H0,35X,45H***FCLOUD ERROR*** TOO MANY INPUT CARDS OR GO,  FCL02230
     + 16H SENTINEL ABSENT,/)                                           FCL02240
      IERR=1                                                            FCL02250
      GO TO 200                                                         FCL02260
  360 CONTINUE                                                          FCL02270
  361 CONTINUE                                                          FCL02280
      ISW=IFLG+IFLO                                                     FCL02290
      IF(ISW.EQ.3) ISW=2                                                FCL02300
      IF(IGEOSW.NE.1) GO TO 222                                         FCL02310
      XC=PTS(13)                                                        FCL02320
      YC=PTS(14)                                                        FCL02330
      ZC=PTS(15)                                                        FCL02340
      XR=PTS(4)                                                         FCL02350
      YR=PTS(5)                                                         FCL02360
      ZR=PTS(6)                                                         FCL02370
      XS=PTS(1)                                                         FCL02380
      YS=PTS(2)                                                         FCL02390
      ZS=PTS(3)                                                         FCL02400
  222 CONTINUE                                                          FCL02410
C                                                                       FCL02420
C  ECHO INPUT                                                           FCL02430
C                                                                       FCL02440
      WRITE(IOOUT,1000) XC,YC,ZC,AE,BE,CE,XR,YR,ZR,XS,YS,ZS             FCL02450
      IF (ETA.LT.1.E-20) CALL PFNN(LAMBDA,0.,INDEXP,PFN,ETA)
      WRITE(IOOUT,1100) INDEXP,ETA,LAMBDA,KAPPA,W0,TAUBAR,THETA0,PHI0,  FCL02460
     1RHO,LB0,TMPA,TMPC,LD                                              FCL02470
      IF(ISW.EQ.2) GO TO 15                                             FCL02480
C                                                                       FCL02490
C  ISW .NE. 2 INDICATES PRELIMINARY GEOMETRICAL CALCULATIONS TO         FCL02500
C  BE PERFORMED: COMPUTE INTERSECTIONS (XM,YM,ZM) AND (XN,YN,ZN) OF     FCL02510
C  LINE OF SIGHT WITH CLOUD, SBAR = LENGTH OF PATH FROM SOURCE TO       FCL02520
C  RECEIVER, S = LENGTH OF PATH IN CLOUD, AND TRANS = TRANSMITTANCE     FCL02530
C  THROUGH CLOUD                                                        FCL02540
C                                                                       FCL02550
      L=XS-XR                                                           FCL02560
      M=YS-YR                                                           FCL02570
      N=ZS-ZR                                                           FCL02580
      SBAR=SQRT(L*L+M*M+N*N)                                            FCL02590
      L=L/SBAR                                                          FCL02600
      M=M/SBAR                                                          FCL02610
      N=N/SBAR                                                          FCL02620
      DX=XS-XC                                                          FCL02630
      DY=YS-YC                                                          FCL02640
      DZ=ZS-ZC                                                          FCL02650
      ASQ=AE*AE                                                         FCL02660
      BSQ=BE*BE                                                         FCL02670
      CSQ=CE*CE                                                         FCL02680
      ABSQ=ASQ*BSQ                                                      FCL02690
      ACSQ=ASQ*CSQ                                                      FCL02700
```

```
      BCSQ=BSQ*CSQ                                                    FCL02710
      ABCSQ=ASQ*BSQ*CSQ                                               FCL02720
      A=BCSQ*L*L+ACSQ*M*M+ABSQ*N*N                                    FCL02730
      B=2.0*(L*DX*BCSQ+M*DY*ACSQ+N*DZ*ABSQ)                           FCL02740
      C=BCSQ*DX*DX+ACSQ*DY*DY+ABSQ*DZ*DZ-ABCSQ                        FCL02750
      DISCRM=B*B-4.0*A*C                                              FCL02760
      IF(DISCRM.GE.0.0) GO TO 10                                      FCL02770
      WRITE(IOOUT,1200)                                               FCL02780
      S=0.0                                                           FCL02790
      TRANS=1.0                                                       FCL02800
      LP=0.0                                                          FCL02810
      GO TO 90                                                        FCL02820
   10 SRTDSC=SQRT(DISCRM)                                             FCL02830
      TWOA=2.0*A                                                      FCL02840
      VPLUS=(-B+SRTDSC)/TWOA                                          FCL02850
      VMINUS=(-B-SRTDSC)/TWOA                                         FCL02860
      XM=XS+VPLUS*L                                                   FCL02870
      YM=YS+VPLUS*M                                                   FCL02880
      ZM=ZS+VPLUS*N                                                   FCL02890
      XN=XS+VMINUS*L                                                  FCL02900
      YN=YS+VMINUS*M                                                  FCL02910
      ZN=ZS+VMINUS*N                                                  FCL02920
      S=SQRT((XM-XN)*(XM-XN)+(YM-YN)*(YM-YN)+(ZM-ZN)*(ZM-ZN))         FCL02930
      IF(S.LT.1.E-4) S=0.0                                            FCL02940
   15 TRANS=EXP(-KAPPA*S)                                             FCL02950
      IF(TMPA.LT.-99.0) GO TO 20                                      FCL02960
C                                                                     FCL02970
C     TMPA >= -99.0 SPECIFIES THERMAL CALCULATION                     FCL02980
C                                                                     FCL02990
      RBAR=(AE+BE+CE)/3.0                                             FCL03000
      CALL THRMCL(RBAR,W0,TMPA,TMPC,LAMBDA,KAPPA,TRANS,LP)            FCL03010
      GO TO 90                                                        FCL03020
C                                                                     FCL03030
C     ISW .NE. 1 SPECIFIES NEW PARAMETERS IN COMMON /OPT/;            FCL03040
C     ASSOCIATED PRELIMINARY COMPUTATIONS YIELD EXTRATERRESTRIAL      FCL03050
C     IRRADIANCE E0, COORDINATES (L0,M0,N0) OF UNIT VECTOR            FCL03060
C     POINTING TO SUN, AND TATM = TRANSMITTANCE OF ATMOSPHERE         FCL03070
C     ABOVE CLOUD                                                     FCL03080
C                                                                     FCL03090
   20 IF(ISW.EQ.1) GO TO 30                                           FCL03100
      CALL ILLUM(LAMBDA,LD,E0)                                        FCL03110
      THT0=THETA0/57.2958                                             FCL03120
      PH0=PHI0/57.2958                                                FCL03130
      L0=SIN(THT0)*COS(PH0)                                           FCL03140
      M0=SIN(THT0)*SIN(PH0)                                           FCL03150
      N0=COS(THT0)                                                    FCL03160
      TATM=EXP(-TAUBAR/N0)                                            FCL03170
   30 WRITE(IOOUT,1600) L,M,N,L0,M0,N0                                FCL03180
      IF(INDEXP.GT.-1) GO TO 40                                       FCL03190
C                                                                     FCL03200
C     INDEXP <  0 SPECIFIES MULTIPLE SCATTERING COMPUTATION;          FCL03210
C     COMPUTE TAU AND TAU0 = CLOUD OPTICAL DEPTH AND THICKNESS,       FCL03220
C     AND CALL MSCLD FOR VALUE OF LP = PATH RADIANCE                  FCL03230
C                                                                     FCL03240
      XG=(XM+XN)/2.0                                                  FCL03250
      YG=(YM+YN)/2.0                                                  FCL03260
      ZG=(ZM+ZN)/2.0                                                  FCL03270
      H=CE*SQRT(1.0-(XG-XC)*(XG-XC)/(AE*AE)-(YG-YC)*(YG-YC)/(BE*BE))  FCL03280
      TAU0=2.0*H*KAPPA                                                FCL03290
      H=ZC-ZG+H                                                       FCL03300
      TAU=KAPPA*H                                                     FCL03310
      CALL MSCLD(TAU,TAU0,TRANS,TATM,E0,W0,ETA,RHO,LP)                FCL03320
      GO TO 90                                                        FCL03330
C                                                                     FCL03340
C     INDEXP > -1 SPECIFIES SINGLE SCATTERING COMPUTATION;            FCL03350
C     COMPUTE SCATTERING ANGLE CHI,PHASE FUNCTION VALUE P =           FCL03360
C     PFN(LAMBDA,CHI), GEOMETRICAL PARAMETERS ALPHA, BETA, GAMMA,     FCL03370
C     DELTA, AND EPS, RGRND = GROUND RADIANCE, AND CALL SSCLD         FCL03380
C     TO COMPUTE PATH RADIANCE                                        FCL03390
C                                                                     FCL03400
```

```
   40 THETA= ACOS(N)
      PHI=0.0
      IF(M.NE.0.0.OR.L.NE.0.0) PHI=ATAN2(M,L)
      COSCHI=N*N0+SIN(THETA)*SIN(THT0)*COS(PHI-PH0)
      CHI= ACOS(COSCHI)*57.2958
      CALL PFNN(LAMBDA,CHI,INDEXP,PFN,ETA)
      P=PFN
      RGRND=E0*TATM*N0
      IF(LB0.LT.1.E-10) LB0=RHO*RGRND
      DX=XM-XC
      DY=YM-YC
      DZ=ZM-ZC
      L0S=L0*L0
      M0S=M0*M0
      N0S=N0*N0
      L0M=L0*M-M0*L
      L0N=L0*N-N0*L
      N0M=N0*M-M0*N
      MXLY=M0*DX-L0*DY
      NXLZ=N0*DX-L0*DZ
      MZNY=M0*DZ-N0*DY
      DENOM=BCSQ*L0S+ACSQ*M0S+ABSQ*N0S
      ALPHA=(BCSQ*L*L0+ACSQ*M*M0+ABSQ*N*N0)/DENOM
      BETA=-(BCSQ*L0*DX+ACSQ*M0*DY+ABSQ*N0*DZ)/DENOM
      GAMMA=-ABCSQ*(CSQ*L0M*L0M+BSQ*L0N*L0N+ASQ*N0M*N0M)/(DENOM*DENOM)
      DELTA=CSQ*MXLY*L0M+BSQ*NXLZ*L0N+ASQ*MZNY*N0M
      DELTA=-2.0*ABCSQ*DELTA/(DENOM*DENOM)
      EPS=BCSQ*L0S+ACSQ*M0S+ABSQ*N0S-CSQ*MXLY*MXLY-BSQ*NXLZ*NXLZ
     1-ASQ*MZNY*MZNY
      EPS=ABCSQ*EPS/(DENOM*DENOM)
      IF(EPS.GE.0.0) GO TO 50
      WRITE(IOOUT,1300) EPS
      EPS=0.0
   50 IF(GAMMA.LT.0.0) GO TO 60
      WRITE(IOOUT,1400) GAMMA
      GO TO 200
   60 CALL SSCLD(ALPHA,BETA,GAMMA,DELTA,EPS,S,KAPPA,W0,RGRND,P,LP)
C
C   WRITE RESULTS
C
   90 F=LP/(LB0*TRANS)
      TC=1.0/(1.0+F)
      WRITE(IOOUT,1500) S,TRANS,LP,TC
  200 RETURN
 1000 FORMAT(1H0,43X,45H-- RADIATIVE TRANSFER THROUGH FINITE CLOUD --/
     1 1H0/45X,14H(XC,YC,ZC) = (,2(F8.4,1H,),F8.4,12H) KILOMETERS/
     2 45X,14H(AE,BE,CE) = (,2(F8.4,1H,),F8.4,1H)/
     3 45X,14H(XR,YR,ZR) = (,2(F8.4,1H,),F8.4,1H)/
     4 45X,14H(XS,YS,ZS) = (,2(F8.4,1H,),F8.4,1H))
 1100 FORMAT(1H0,45X,8HINDEXP =,I9,10X,8HETA     =,F11.3/
     1 45X,8HLAMBDA =,F9.3,8H (MU)    ,2X,9HKAPPA   = ,1PE10.4,7H (KM-1)/
     2 45X,8HOMEGA 0=,0PF9.3,10X,8HTAUBAR =,1X,1PE10.4/
     3 45X,8HTHETA0 =,0PF9.1,10X,8HPHI0    =,F11.1,10H (DEGREES)/
     4 45X,8HRHO    =,0PF9.3,10X,8HLB0     =,F11.3,13H (W/M2-SR-MU)/
     5 45X,8HTMPA   =,F9.1,8H (DEG.C),2X,8HTMPC    =,F11.1/
     6 45X,8HLD     =,I9)
 1200 FORMAT(1H0,44X,43H**LINE-OF-SIGHT MISSES CLOUD.  S SET TO 0.0)
 1300 FORMAT(1H0,44X,4HEPS=,E10.4,24H LT 0.0.  EPS SET TO 0.0)
 1400 FORMAT(1H0,44X,6HGAMMA=,E10.4,29H  GE 0.0.  SKIP TO NEXT CASE.)
 1500 FORMAT(1H0/1H0,37X,11HPATH LENGTH,3X,15HTRANSMITTANCE   ,
     1 13HPATH RADIANCE,4X,8HCONTRAST/
     2 38X,11H(IN CLOUD) ,19X,12H(W/M2-SR-MU),15H  TRANSMITTANCE/
     3 36X,4(15H+-------------+)/
     4 1H0,40X,F6.3,7X,1PE9.3,5X,1PE9.3,5X,1PE9.3)
 1600 FORMAT(1H0,45X,28HUNIT SOURCE VECTOR L, M, N =,3(1X,F7.4)/
     1 1H0,45X,23HSOLAR VECTOR L0,M0,N0 =,3(1X,F7.4))
      END
```

```
      SUBROUTINE THRMCL(RBAR,W0,TMPA,TMPC,LAMBDA,KAPPA,TRANS,LP)
      REAL LAMBDA,LP,LI,KAPPA
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      BB(X,T)=1.19106E8/(X**5*(EXP(1.4388E4/(X*T))-1.0))
C*********************************************************************
C
C  SUBROUTINE PERFORMS THERMAL RADIATION CALCULATIONS FOR FINITE
C  CLOUD, RETURNING THE VALUE OF LP = PATH RADIANCE.
C  INPUTS ARE:
C
C     RBAR = AVERAGE HALF-LENGTH OF PATH THROUGH CLOUD
C     W0 = SINGLE SCATTERING ALBEDO
C     TMPA = TEMPERATURE OF THE ATMOSPHERE
C     TMPC = TEMPERATURE OF THE CLOUD
C     LAMBDA = WAVELENGTH
C     KAPPA = VOLUME EXTINCTION COEFFICIENT
C     TRANS = TRANSMITTANCE THROUGH CLOUD.
C
C*********************************************************************
      G=1.0-EXP(-KAPPA*RBAR)
      B=BB(LAMBDA,273.16+TMPC)
      LI=BB(LAMBDA,273.16+TMPA)
      WRITE(IOOUT,1000) B,LI
      LP=(1.0-TRANS)*((1.0-W0)*(1.0+G*W0)*B+W0*(1.0-G)*LI)
 1000 FORMAT(1H0,43X,40H  **THERMAL CALCULATION OF PATH RADIANCE/,45X,
     122H    BB(LAMBDA,TMPC) = ,1PE10.4,11H W/M2-SR-MU/,45X,
     122H    BB(LAMBDA,TMPA) = ,1PE10.4,11H W/M2-SR-MU)
      RETURN
      END
```

```
      SUBROUTINE SSCLD(ALPHA,BETA,GAMMA,DELTA,EPS,S,KAPPA,
     1W0,RGRND,P,LP)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C***********************************************************************
C
C  SUBROUTINE COMPUTES PATH RADIANCE LP FOR THE CASE OF SINGLE
C  SCATTERING IN A FINITE CLOUD.  INPUTS ARE GEOMETRICAL PARAMETERS
C  ALPHA, BETA, GAMMA, DELTA, AND EPS, S = LENGTH OF OPTICAL PATH
C  THROUGH CLOUD, AND:
C      KAPPA = VOLUME EXTINCTION COEFFICIENT
C      W0 = SINGLE SCATTERING ALBEDO
C      RGRND = SURFACE BACKGROUND RADIANCE
C      P = SINGLE SCATTERING PHASE FUNCTION.
C
C***********************************************************************
      REAL KAPPA,LP
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      ASIN(ARG)=ATAN2(ARG,SQRT(1.-ARG**2))
      WRITE(IOOUT,1500)
      DISCRM=DELTA*DELTA-4.0*GAMMA*EPS
      IF(DISCRM.GT.0.0) GO TO 10
      WRITE(IOOUT,1000) DISCRM
      LP=0.0
      GO TO 20
   10 TGSDEL=2.0*GAMMA*S+DELTA
      HS= ASIN(TGSDEL/SQRT(DISCRM))/SQRT(-GAMMA)
      H0= ASIN(DELTA/SQRT(DISCRM))/SQRT(-GAMMA)
      TRANS=EXP(-KAPPA*S)
      BETAB=W0*KAPPA
      LP=(1.0-BETA*KAPPA)*S+(1.0-ALPHA)*KAPPA*S*S/2.0
      LP=LP-KAPPA*DISCRM*(HS-H0)/(8.0*GAMMA)
      LP=LP-KAPPA*(TGSDEL*SQRT(GAMMA*S*S+DELTA*S+EPS)-
     1DELTA*SQRT(EPS))/(4.0*GAMMA)
      LP=BETAB*LP*P*RGRND*TRANS/(4.0*PI)
   20 RETURN
 1000 FORMAT(9H0DISCRM =,E10.4,25H IN SSCLD.  LP SET TO 0.0)
 1500 FORMAT(1H0,43X,33H   **RESULTS FOR SINGLE SCATTERING)
      END
```

```
      SUBROUTINE PFNN(LAMBDA,CHI,INDEXP,PFN,ETA)                        PFN00010
      REAL LAMBDA,LAMDA1,LAM1,LAM2,KAPPA                                PFN00020
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUPFN00030
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     PFN00040
      DIMENSION PHANG(65),PF1(65),PF2(65)                               PFN00050
      DATA LAMDA1,IP1,CHI1/-1.0,-2,-1.0/                                PFN00060
C***********************************************************************PFN00070
C                                                                       PFN00080
C  SUBROUTINE COMPUTES THE VALUE OF THE INDEXP-TH PHASE FUNCTION        PFN00090
C  AT SCATTERING ANGLE CHI AND WAVELENGTH LAMBDA USING BILINEAR         PFN00100
C  INTERPOLATION.                                                       PFN00110
C                                                                       PFN00120
C***********************************************************************PFN00130
      ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)
      IERR=0                                                            PFN00140
      MAXID=12
C     PFNDAT ONLY CONTAINS ONE PFN FOR THE VISIBLE (.55UM) AND
C     THE NEAR IR (1.06UM); THEREFORE DO NOT INTERPOLATE.
      IWAVE=0
      IF (LAMBDA.LT.2.0) IWAVE=1
C     PRELIMINARIES FOR FINDING ETA
      ETA1=0.0
      ETA2=0.0
      IDCK=0
      IF (INDEXP.LT.0) IDCK=1
      IF (INDEXP.LT.0) INDEXP=-INDEXP
      IF(INDEXP.EQ.IP1.AND.CHI.EQ.CHI1.AND.LAMBDA.EQ.LAMDA1)            PFN00150
     1GO TO 200                                                         PFN00160
      IF(INDEXP.EQ.IP1.AND.LAMBDA.EQ.LAMDA1) GO TO 70                   PFN00170
      REWIND IPHFUN                                                     PFN00180
      NRD=0                                                             PFN00190
      DO 20 I=1,66,11                                                   PFN00200
      READ(IPHFUN,1000) (PHANG(I+J-1),J=1,11)                           PFN00210
      DO 10 J=1,11                                                      PFN00220
      NRD=NRD+1                                                         PFN00230
      IF(PHANG(NRD).GE.999.9) GO TO 30                                  PFN00240
   10 CONTINUE                                                          PFN00250
   20 CONTINUE                                                          PFN00260
   30 NA=NRD-1                                                          PFN00270
      DO 35 I=1,NA                                                      PFN00280
35    PHANG(I)=COS(PHANG(I)*PIRAD)                                      PFN00290
40    CONTINUE                                                          PFN00300
C     END-OF-FILE CHECK                                                 PFN00310
      IF (IERR.EQ.2) GO TO 195                                          PFN00320
      READ(IPHFUN,1100) IANG1,ID,LAM1,W0,KAPPA,BETA                     PFN00330
      IF (LAM1.GE.12.00.AND.ID.EQ.MAXID) IERR=2                         PFN00340
      READ(IPHFUN,1200) (PF1(I),I=1,NA)                                 PFN00350
      SUM=0.                                                            PFN00360
C     START RENORMALIZATION OF PHASE FUNCTION - ALSO SEE BELOW          PFN00370
      DO 45 J=2,NA                                                      PFN00380
45    SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF1(J)+PF1(J-1))/4.               PFN00390
      DO 46 J=1,NA                                                      PFN00400
46    PF1(J)=PF1(J)/SUM                                                 PFN00410
      ETA1=ETAINT(PF1,PHANG,NA)
      IF(ID.NE.INDEXP) GO TO 40                                         PFN00420
      IF (IWAVE.EQ.1.AND.LAMBDA.GT.LAM1)  GO TO 40
      IF (IWAVE.EQ.1) GO TO 75
      IF(LAMBDA.LT.LAM1) GO TO 190                                      PFN00430
50    CONTINUE                                                          PFN00440
      IF (IERR.EQ.2) GO TO 195                                          PFN00450
      READ(IPHFUN,1100) IANG2,ID,LAM2,W0,KAPPA,BETA                     PFN00460
      IF (LAM2.GE.12..AND.ID.EQ.MAXID) IERR=2                           PFN00470
      READ(IPHFUN,1200) (PF2(I),I=1,NA)                                 PFN00480
      SUM=0.                                                            PFN00490
      DO 55 J=2,NA                                                      PFN00500
55    SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF2(J)+PF2(J-1))/4.               PFN00510
      DO 56 J=1,NA                                                      PFN00520
56    PF2(J)=PF2(J)/SUM                                                 PFN00530
      ETA2=ETAINT(PF2,PHANG,NA)
C     THE PHASE FUNCTION(S) ARE NOW NORMALIZED TO: INTEGRAL OF          PFN00540
```

```
C     PHASE FUNCTION OVER ALL SOLID ANGLE DIVIDED BY 4 PI = 1.               PFN00550
      IF(ID.NE.INDEXP) GO TO 190                                             PFN00560
      IF(LAMBDA.LE.LAM2) GO TO 70                                            PFN00570
      LAM1=LAM2                                                              PFN00580
      IANG1=IANG2                                                            PFN00590
      DO 60 I=1,NA                                                           PFN00600
      PF1(I)=PF2(I)                                                          PFN00610
   60 CONTINUE                                                               PFN00620
      GO TO 50                                                               PFN00630
70    CONTINUE
      IF (IWAVE.EQ.1) GO TO 75
      DLAM=(LAMBDA-LAM1)/(LAM2-LAM1)                                         PFN00640
75    CONTINUE
      IF(CHI.LT.-1.E-3.OR.CHI.GT.180.001) GO TO 190                          PFN00650
      DO 80 J=2,NA                                                           PFN00670
      IF(CHI.LE.(ACOS(PHANG(J))/PIRAD)) GO TO 90                             PFN00680
   80 CONTINUE                                                               PFN00690
      J=NA                                                                   PFN00700
   90 J1=J-1                                                                 PFN00710
      DCHI=(COS(CHI*PIRAD)-PHANG(J1))/(PHANG(J)-PHANG(J1))                   PFN00720
      IF (IWAVE.NE.1) GO TO 95
      PFN=PF1(J1)+DCHI*(PF1(J)-PF1(J1))
      GO TO 96
95    PFN=PF1(J1)+DLAM*(PF2(J1)-PF1(J1))+DCHI*(PF1(J)-PF1(J1))               PFN00730
     1+DLAM*DCHI*(PF2(J)+PF1(J1)-PF2(J1)-PF1(J))                             PFN00740
96    LAMDA1=LAMBDA                                                          PFN00760
      CHI1=CHI                                                               PFN00770
      IF (IP1.NE.-2) WRITE(IOOUT,1500) CHI,PFN
      IP1=INDEXP
      ETA=ETA1+DLAM*(ETA2-ETA1)
      GO TO 200                                                              PFN00790
  190 WRITE(IOOUT,1300) ID,INDEXP,LAM1,LAMBDA,CHI                            PFN00800
      STOP                                                                   PFN00810
195   WRITE (IOOUT,1600) IPHFUN                                              PFN00820
      STOP                                                                   PFN00830
200   CONTINUE
      IF (IDCK.EQ.1) INDEXP=-INDEXP
      RETURN                                                                 PFN00840
 1000 FORMAT(11(F6.2,1X))                                                    PFN00850
 1100 FORMAT(2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))                           PFN00860
 1200 FORMAT(6(E12.6,1X))                                                    PFN00870
 1300 FORMAT(33H0ERROR IN READING PHASE FUNCTION./                           PFN00880
     127H ID,INDEXP,LAM1,LAMBDA,CHI=,2I3,3E13.7)                             PFN00890
 1500 FORMAT(1H0,23HSCATTERING ANGLE CHI = ,F8.2,7X,24H PHASE FN P(LAMBDPFN00900
     +A,CHI)=,E10.4)                                                         PFN00910
1600  FORMAT(1X,32HATTEMPT TO READ PAST EOF ON UNIT,I3,18H IN SUBROUTINEPFN00920
     1 PFN//)                                                                PFN00930
      END                                                                    PFN00940
```

```
      SUBROUTINE MSCLD(TAU,TAU0,TRANS,TATM,E0,W0,ETA,RHO,LP)
      REAL LP,LBAR,K
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      SINH(ARG)=.5*(EXP(ARG)-EXP(-ARG))
      COSH(ARG)=.5*(EXP(ARG)+EXP(-ARG))
C***********************************************************************
C
C   SUBROUTINE COMPUTES LP = PATH RADIANCE DUE TO MULTIPLE SCATTERING
C   IN A FINITE CLOUD.  INPUTS ARE:
C
C      TAU = CLOUD OPTICAL DEPTH
C      TAU0 = CLOUD OPTICAL THICKNESS
C      TRANS = TRANSMITTANCE ALONG LINE OF SIGHT
C      TATM = TRANSMITTANCE OF ATMMOSPHERE ABOVE CLOUD
C      E0 = EXTRATERRESTRIAL IRRADIANCE
C      W0 = SINGLE SCATTERING ALBEDO
C      ETA = FORWARD SCATTERING PARAMETER
C      RHO = BACKGROUND SURFACE REFLECTANCE.
C
C***********************************************************************
      IF(W0.LT.1.0) GO TO 10
      W0=.999
      WRITE(IOOUT,2000)
   10 C1=1.0-ETA*W0
      C2=(1.0-ETA)*W0
      K=SQRT((1.0-W0)*(1.0+W0-2.0*ETA*W0))
      TOK=K*TAU0
      TOTK=K*(TAU0-TAU)
      TK2=2.0*K*TAU
      GAMMA=E0*TATM/(C1*SINH(TOK)+K*COSH(TOK))
      DELTA=RHO*K/((C1-RHO*C2)*SINH(2.0*TOK)+K*COSH(2.0*TOK))
      WRITE(IOOUT,1000)
      EPLUS=C2*SINH(TOTK)+DELTA*(C1*SINH(TK2)+K*COSH(TK2))
      EMINUS=C1*SINH(TOTK)+K*COSH(TOTK)+DELTA*C2*SINH(TK2)
      LBAR=GAMMA*(EPLUS+EMINUS)/TWOPI
      LP=W0*LBAR*(1.0-TRANS)
      WRITE(IOOUT,100)TAU,TAU0,TATM
  100 FORMAT(1H0,45X,18H  OPTICAL DEPTH = ,1PE10.4/,45X,
     131H  OPTICAL THICKNESS OF CLOUD = ,1PE10.4/,45X,
     244H  TRANSMITTANCE OF ATMOSPHERE ABOVE CLOUD = ,1PE10.4)
      RETURN
 1000 FORMAT(36H0  **RESULTS FOR MULTIPLE SCATTERING )
 2000 FORMAT(47H0  **OMEGA 0 WAS 1.0, NOW SET TO 0.999 IN MSCLD)
      END
```

```
      FUNCTION ETAINT(PFN,PHANG,NA)
C     THIS FUNCTION WILL DETERMINE ETA, THE FORWARD SCATTERING
C     PARAMETER: ETA=.5*INTEGRAL PFN OVER THETA, WHERE THETA GOES
C     FROM ZERO TO PI/2.
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DIMENSION PHANG(65),PFN(65)
      NAM1=NA-1
      ETA=0.
      DO 1 I=1,NAM1
      IF(PHANG(I+1).GT.0.) ETA=ETA+(PHANG(I)-PHANG(I+1))*
     1 (PFN(I+1)+PFN(I))/4.
1     CONTINUE
      ETAINT=ETA
      RETURN
      END
```

```
      SUBROUTINE OVRCST(LAMBDA,TRANS,IERR)
      REAL LC,LG,LB0,KAPPA,LAMBDA,MU,LP
      DIMENSION IALPH(7)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /GEOMET/PTS(15),IGEOSW
      DATA IZERO/0/
      DATA IALPH/2HOP  ,2HSP  ,2HCL  ,2HBK  ,2HGR  ,2HTE  ,2HGO  /
      BB(X,T)=1.191062E8/(X**5*(EXP(1.4387864E4/(X*T))-1.0))
C*********************************************************************
C
C   SUBROUTINE COMPUTES BEAM TRANSMITTANCE, PATH RADIANCE, AND
C   CONTRAST TRANSMITTANCE  ALONG AN OPTICAL PATH UNDER AN OVERCAST
C   SKY.  ORDER-INDEPENDENT INPUT CARDS ARE AS FOLLOWS:
C   (INDIVIDUAL RECORD FORMAT IS (A4,1X,5(E10.4,1X))
C
C    MNEMONIC       VARIABLES READ           DESCRIPTION
C    --------       --------------           -----------
C
C     OPOS              XO,YO,ZO            OBSERVER POSITION
C     SPOS              XT,YT,ZT            SOURCE POSITION
C     CLDS        ZC,LC,KAPPA,ETA ,W0       CLOUD PARAMETERS
C     BKGR              LB0                 BACKGROUND RADIANCE
C     GRND              LG                  GROUND RADIANCE
C     TEMP              TEMP                TEMPERATURE ALONG PATH
C     GO                                    END OF READ SENTINEL
C
C      THE VARIABLES ZC AND LC REFER TO THE OVERCAST SKY LAYER!
C      THE VARIABLES KAPPA, ETA, W0, TEMP, REFER TO THE ATMOSPHERE
C      BETWEEN THE OVERCAST SKY AND GROUND I.E. THE INTERVENING
C      ATMOSPHERIC PROPERTIES (GAS OR AEROSOL)
C  ** NOTE : THE GO CARD MUST BE THE LAST RECORD READ.
C
C   THE FOLLOWING ENUMERATES THE VARIABLES LISTED ON THE ABOVE CARDS :
C
C        (XO,YO,ZO) = OBSERVER COORDINATES
C        (XT,YT,ZT) = SOURCE COORDINATES
C        ZC = HEIGHT OF CLOUD LAYER
C        LC = CLOUD RADIANCE
C        LG = GROUND RADIANCE
C        LB0 = BACKGROUND RADIANCE
C        KAPPA = VOLUME EXTINCTION COEFFICIENT (KM-1)
C        ETA   = FORWARD SCATTERING PARAMETER
C        W0 = SINGLE SCATTERING ALBEDO
C        TEMP = TEMPERATURE ALONG PATH (DEG. C)
C
C   LENGTH UNITS ARE KILOMETERS; RADIANCE UNITS ARE W/M2-SR-MU.
C   IF TEMP >= -99., THERMAL RADIATION IS CALCULATED; IF TEMP < -99.
C   SINGLY SCATTERED RADIATION IS CALCULATED.
C
C   SUBROUTINE RETURNS:
C      TRANS = BEAM TRANSMITTANCE
C   TO CALLING PROGRAM.
C
C*********************************************************************
C
C   DATA INITIALIZATION
C
      IF(IZERO.NE.0) GO TO 477
      XO=0.0
      YO=0.0
      ZO=0.0
      XT=0.0
      YT=0.0
      ZT=0.0
      ZC=0.0
      LC=0.0
      KAPPA=0.0
      ETA =0.0
      W0=0.0
      LB0=0.0
```

```
      LG=0.0
      TEMP=0.0
      IZERO=1
  477 CONTINUE
      DO 360 K=1,7
      READ(IOIN,334)IA,IA2,R1,R2,R3,R4,R5
  334 FORMAT(2A2,1X,5(E10.4,1X))
      DO 333 I=1,8
      IF(IA.NE.IALPH(I)) GO TO 333
      IND=I
      IF(IND.EQ.7) GO TO 361
  333 CONTINUE
      IF(IND.EQ.8) GO TO 355
      IF(K.EQ.7.AND.IND.NE.7) GO TO 358
      GO TO (341,342,343,344,345,346),IND
  341 XO=R1
      YO=R2
      ZO=R3
      GO TO 360
  342 XT=R1
      YT=R2
      ZT=R3
      GO TO 360
  343 ZC=R1
      LC=R2
      KAPPA=R3
      ETA =R4
      WO=R5
      GO TO 360
  344 LB0=R1
      GO TO 360
  345 LG=R1
      GO TO 360
  346 TEMP=R1
      GO TO 360
  355 WRITE(IOOUT,357)
  357 FORMAT(1H0,25X,44H***OVRCST ERROR*** INPUT CARD DETECTED WHICH,
     + 36H DOES NOT MATCH CORRECT INPUT FORMAT,/)
      GO TO 360
  358 WRITE(IOOUT,359)
  359 FORMAT(1H0,34X,45H***OVRCST ERROR*** TOO MANY INPUT CARDS OR GO,
     + 16H SENTINEL ABSENT,/)
      IERR=1
      GO TO 200
  360 CONTINUE
  361 CONTINUE
      IF(IGEOSW.NE.1) GO TO 222
      XO=PTS(4)
      YO=PTS(5)
      ZO=PTS(6)
      XT=PTS(1)
      YT=PTS(2)
      ZT=PTS(3)
  222 CONTINUE
C
C  ECHO INPUT
C
      WRITE(IOOUT,1000) XO,XT,YO,YT,ZO,ZT,ZC,LC,LAMBDA,LG,
     1TEMP,LB0,KAPPA,WO,ETA
      IF(TEMP.LT.-99.0) GO TO 4
      BBTEMP=BB(LAMBDA,273.16+TEMP)
      WRITE(IOOUT,1800) BBTEMP
      BTE=(WO-1.0)*BBTEMP
      GO TO 8
    4 WRITE(IOOUT,1700)
    8 F=2.0*(1.0-ETA )
      ZLEN=ZT-ZO
      S=SQRT((XT-XO)**2+(YT-YO)**2+ZLEN**2)
      MU=ABS(ZLEN)/S
      TO=KAPPA*ZO
```

```
      TT=KAPPA*ZT
      SO=KAPPA*(ZC-ZO)
      ST=KAPPA*(ZC-ZT)
      TRANS=EXP(-KAPPA*S)
      CONST=W0*ETA /2.0
      IF(ZLEN) 10,40,70
C
C  HEIGHT OF OBSERVER > HEIGHT OF SOURCE
C
   10 G21=G2(TO,TT,MU)
      G22=G2(SO,ST,-MU)
      T1=EXP(-TO/MU)
      T2=EXP(SO/MU)
      F1=W0*F*KAPPA*S*T1/2.0
      XINT1=CONST*T1*G21
      XINT2=CONST*T2*G22
      IF(TEMP.GE.-99.0) GO TO 20
      LP=LG*(F1+XINT1)+LC*XINT2
      GO TO 100
   20 LP=(LG+BTE)*(F1+XINT1)+(LC+BTE)*XINT2+W0*BTE*(TRANS-1.0)
      GO TO 100
C
C  HEIGHT OF OBSERVER = HEIGHT OF SOURCE
C
   40 E21=1.0
      E22=1.0
      IF(TO.NE.0.0) E21=EXP(-TO)-TO*E1(TO)
      IF(SO.NE.0.0) E22=EXP(-SO)-SO*E1(SO)
      IF(TEMP.GE.-99.0) GO TO 50
      LP=CONST*(1.0-TRANS)*(LG*E21+LC*E22)
      GO TO 100
   50 LP=CONST*(E21*(LG+BTE)+E22*(LC+BTE))
      LP=(LP-W0*BTE)*(1.0-TRANS)
      GO TO 100
C
C  HEIGHT OF OBSERVER < HEIGHT OF SOURCE
C
   70 G21=G2(TO,TT,-MU)
      G22=G2(SO,ST,MU)
      T1=EXP(-SO/MU)
      T2=EXP(TO/MU)
      F1=W0*F*KAPPA*S*T1/2.0
      XINT1=CONST*T2*G21
      XINT2=CONST*T1*G22
      IF(TEMP.GE.-99.0) GO TO 80
      LP=LC*(F1+XINT2)+LG*XINT1
      GO TO 100
   80 LP=(LC+BTE)*(F1+XINT2)+(LG+BTE)*XINT1+W0*BTE*(TRANS-1.0)
C
C  WRITE RESULTS
C
  100 TC=1.0/(1.0+LP/(LB0*TRANS))
      WRITE(IOOUT,2000) S,TRANS,LP,TC
  200 RETURN
 1000 FORMAT(1H0/1H0,55X,34H-- RADIATION UNDER OVERCAST SKY --/
     1 1H0,43X,8HXO      =,5X,F6.3,5H (KM),8X,5HXT  =,5X,F6.3,5H (KM)/
     2 44X,8HYO      =,5X,F6.3,13X,5HYT  =,5X,F6.3/
     3 44X,8HZO      =,5X,F6.3,13X,5HZT  =,5X,F6.3/
     4 44X,8HZC      =,5X,F6.3,13X,5HLC  =,1X,1PE10.4,13H (W/M2-SR-MU)/
     5 44X,8HLAMBDA =,5X,0PF6.3,5H (MU),8X,5HLG  =,1X,1PE10.4/
     6 44X,8HTEMP    =,3X,0PF6.1,10H   (DEG.C),5X,5HLB0 =,1X,1PE10.4/
     7 44X,8HKAPPA   =,1X,E10.4,7H (KM-1),6X,5HW0  =,5X,0PF6.3/
     8 44X,8HETA     =,6X,0PF5.3)
 2000 FORMAT(1H0,39X,4HPATH,7X,13HTRANSMITTANCE,2X,13HPATH RADIANCE,
     1 4X,8HCONTRAST/
     2 38X,11HLENGTH (KM),19X,12H(W/M2-SR-MU),15H  TRANSMITTANCE/
     3 36X,4(15H+-------------+)/
     4 1H0,39X,F6.3,9X,F7.5,7X,1PE9.3,7X,0PF7.5)
 1700 FORMAT(1H0,50X,31H**RESULTS FOR SINGLE SCATTERING)
 1800 FORMAT(1H0,46X,38H**THERMAL CALCULATION OF PATH RADIANCE/
```

```
1 1H0,50X,8HBBTEMP= ,1PE10.4,11H W/M2-SR-MU)                              OVR02070
 END                                                                       OVR02080
```

```
      FUNCTION G2(TAU1,TAU2,MU)
C*********************************************************************
C
C   SUBROUTINE COMPUTES IN CLOSED FORM INTEGRALS OF THE FUNCTION
C
C      EXP(TAU/MU)*E2(TAU)
C
C   WHERE E2 IS THE SECOND EXPONENTIAL INTEGRAL.  FOR DETAILS SEE
C   KOURGANOFF, 'BASIC METHODS IN TRANSFER PROBLEMS', APPENDIX I
C   (PAGES 256-257 OF FIRST EDITION, 1952, OXFORD UNIVERSITY PRESS.)
C
C*********************************************************************
      REAL MU
      DATA GAMMA/.5772156649/
      IF(MU.LT..9999) GO TO 50
C
C   MU EQ 1.0
C
      IF(TAU1.NE.0.0) GO TO 10
      G2=(TAU2-1.0)*EXP(TAU2)*E1(TAU2)-GAMMA-ALOG(TAU2)
      GO TO 100
   10 IF(TAU2.NE.0.0) GO TO 20
      G2=GAMMA+ALOG(TAU1)+EXP(TAU1)*(1.0-TAU1)*E1(TAU1)
      GO TO 100
   20 G2=EXP(TAU1)*(1.0-TAU1)*E1(TAU1)-EXP(TAU2)*(1.0-TAU2)*E1(TAU2)
      G2=G2+(ALOG(TAU1)-ALOG(TAU2))
      GO TO 100
C
C   MU NE 1.0
C
   50 RM=1.0-1.0/MU
      IF(TAU1.NE.0.0) GO TO 60
      G2=EXP(TAU2/MU)*(TAU2-MU)*E1(TAU2)
      G2=G2+MU*E1(TAU2*RM)
      G2=G2+1.0+MU*ALOG(ABS(RM))-EXP(-RM*TAU2)
      GO TO 100
   60 IF(TAU2.NE.0.0) GO TO 70
      G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1)
      G2=G2-MU*E1(TAU1*RM)
      G2=G2+EXP(-RM*TAU1)-MU*ALOG(ABS(RM))-1.0
      GO TO 100
   70 G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1)
      G2=G2-EXP(TAU2/MU)*(MU-TAU2)*E1(TAU2)
      G2=G2+(EXP(-TAU1*RM)-EXP(-TAU2*RM))
      G2=G2+MU*(E1(TAU2*RM)-E1(TAU1*RM))
  100 RETURN
      END
```

```
      FUNCTION E1(X)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C***********************************************************************
C
C   FUNCTION COMPUTES THE VALUE OF THE FIRST EXPONENTIAL INTEGRAL
C   -EI(-X) WHERE -174 < X < 170.  FOR X OUTSIDE THESE BOUNDS, AN
C   OVERFLOW OR UNDERFLOW MIGHT OCCUR, SO PROGRAM EXECUTION IS
C   HALTED.  POLYNOMIAL AND RATIONAL FUNCTION APPROXIMATIONS ARE
C   ADAPTED FROM THE IBM SCIENTIFIC SUBROUTINE PACKAGE, SUBROUTINE
C   EXPI.
C
C***********************************************************************
      DATA GAMMA/.5772156649/
      DATA C1,C2,C3,C4,C5,C6,C7,C8/674.567029,57.411833,6.05529232,
     11699.06552,841.654932,49.3133893,8.01957683,.99979204/
      DATA D1,D2,D3,D4,D5,D6,D7,D8,D9/248.6697,224.4234,32.43665,
     13.061037,.05176245,180.7837,22.63818,38.93944,3.995161/
      DATA F1,F2,F3,F4,F5,F6,F7,F8,F9/9.999999E-1,2.500001E-1,
     15.555682E-2,1.041576E-2,1.664156E-3,2.335379E-4,2.928433E-5,
     21.766345E-6,7.122452E-7/
      DATA G1,G2,G3,G4,G5,G6,G7,G8/.2677737343,8.6347608925,
     118.059016973,8.5733287401,3.9584969228,21.0996530827,
     225.6329561486,9.5733223454/
      IF(X.GT.-174.0.AND.X.LE.170.0) GO TO 10
      WRITE(IOOUT,1000)
      STOP
   10 IF(X.GT.-9.0) GO TO 20
      E1=1.0-(C1+C2*X-C3*X*X-X*X*X)/(C4+C5*X+C6*X*X-C7*X*X*X-C8*X*X*X*X)
      E1=E1*EXP(-X)/X
      GO TO 100
   20 IF(X.GT.-3.0) GO TO 30
      E1=D1+D2*X+D3*X*X+D4*X*X*X+D5*X*X*X*X
      E1=(1.0-E1/(D6+D7*X+D8*X*X+D9*X*X*X+X*X*X*X))*EXP(-X)/X
      GO TO 100
   30 IF(X.GT.1.0) GO TO 40
      E1=F1-X*(F2-X*(F3-X*(F4-X*(F5-X*(F6-X*(F7-X*(F8-X*F9)))))))
      E1=X*E1-GAMMA-ALOG(ABS(X))
      GO TO 100
   40 E1=(G1+X*(G2+X*(G3+X*(G4+X))))/(G5+X*(G6+X*(G7+X*(G8+X))))
      E1=E1*EXP(-X)/X
  100 RETURN
 1000 FORMAT(4H0X =,3X,E10.4,39H OUT-OF-RANGE FOR E1. EXECUTION HALTED.)
      END
```

```
      SUBROUTINE GRNADE(WAVE1,ICLMAT,TRANS,IERR)
C/*********************************************************************/
C/*                    SUBROUTINE GRNADE                              *
C/*                  MAIN GRNADE MODULE                               *
C/*                      EOSAEL80                                    */
C/*********************************************************************/
      DIMENSION XA(3),XTRAN(7)
      COMMON /MO5/ XDTA(1000),CDTA(1000),CL(1000)
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDMAT,CLDHYT,FOGPRB,
     *                 WNDVEL,WNDDIR,IPASCT
      COMMON/MECH0/XM,YM,ZM,XO,YO,ZO,XT,YT,ZT,ISTO,IETO,IDTO,XN,FW,
     *TBURN,ITYPE,EFF,YF,RHA,UW,WD,ICAT,AIRT,TGRAD,BRATE,HEAD,RNG,
     *DLEN,WPOWR,EXTC(8),XMIS(8),XNORTH
      COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
      COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH
      COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM
C**********************************************************************
C** THE FOLLOWING VARIABLES SUPPLIED BY THE USER.
C**  FIELD DATA
C**      XNORTH     FIELD COORDINATES FROM NORTH (DEGREES)
C**      HEAD       GRENADE TANK HEADING CLOCKWISE FROM NORTH (DEGREES)
C**      RNG        DISTANCE OF GRENADES FROM TANK (METERS)
C**      DLEN       GRENADE SPACING (PERPENDICILAR TO HEADING)
C**      XO,YO,ZO   COORDINATES OF OBSERVER (METERS)
C**      XM,YM,ZM   COORDINATES OF TANK (METERS)
C**      XT,YT,ZT   TARGET COORDINATES (METERS)
C** METEOROLOGICAL DATA
C**      WINDP      WIND PROFILE EXPONENT
C**      HM         HEIGHT OF INVERSION LAYER
C**      WD         WIND DIRECTION FROM NORTH (DEGREES)
C**      WS         WIND SPEED (METERS PER SECOND)
C**      RH         RELATIVE HUMIDITY
C**      ICAT       PASQUILL CATEGORY
C**      YF         SMOKE YIELD FACTOR
C** MUNITION DATA
C**      EFF        CLOUD-MAKING EFFIENCY OF MUNITION
C**      QMUN       TOTAL MASS OF SMOKE AGENT (GRAMS)
C** DETECTOR DATA
C**      WAVE1      WAVELENGTH OF INTEREST (MICRONS)
C** DIFFUSION PARAMETERS
C**      SIGZ       REFERENCE SIGMA (METERS)
C**      XREF       REFERENCE DISTANCE (METERS)
C**      ZDIFF      VERTICAL DIFFUSION CONSTANT
C**      YDIFF      CROSSWIND DIFFUSION CONSTANT
C**      HK         TERRAIN SCAVENGING COEFFICIENT
C**      VS         PARTICLE SETTLING VELOCITY (CM/SEC)
C**      RC         TERRAIN REFLECTION COEFFICIENT
C**********************************************************************
C**    DEFINITIONS OF OTHER VARIABLES***
C**      IXMAX      NUMBER OF POINTS ALONG LINE-OF-SIGHT FOR CL COMPUTAIO
C**      NBPT       NUMBER OF GRENADE LINES (NBPT=1)
C**      NTARG      NUMBER OF TARGETS (NTARG=1)
C**      TBURST     MUNITION DETONATION TIME
C*********************************************************************
C**    ERROR CODES AND OPTION CODES;
C**      IWRIT=1    DEPRESSES RAW DATA PRINTOUT
C**      IFLAG=4    INVALAD DATA CARD (BUT IGNORED)
C**      IFLAG=3    OVER 11 DATA CARDS ENTERED BEFORE GO
C**                 REMAINDER IGNORED
C**      IFLAG=2    NORMAL READ TERMINATION
C**      IFLAG=1    WAVELENGTH OF INTEREST NOT IN DEFINED BANDS
C**                 TRANS SET TO 1.0
C*****SET DEFAULTS*******
      TBURST=0.0
      CALL GOGET(WAVE1,KWAVE)
C*****READ DATA AND WRITE HEADING*****
      WRITE(IOOUT,8000)
 8000 FORMAT(1H0,20X,40(2H**),/,21X,1H*,34X,14HPROGRAM GRNADE,30X,1H*,/,
```

```
     *21X,1H*,37X,8HEOSAEL80,33X,1H*,/,21X,40(2H**))
      IWRIT=1
      IFLAG=0
    6 CALL DATRD(IWRIT,IFLAG)
      IERR=IFLAG
      IF(IFLAG.EQ.4)GO TO 9999
C******CALCULATE INTEGRATION INCREMENT
      XX=(XT-XO)**2+(YT-YO)**2+(ZT-ZO)**2
      DLOS=SQRT(XX)
      IXMAX=IFIX(DLOS)
      IF(IXMAX.GT.1000)IXMAX=1000
C*****DEFAULT TO CLIMATE DATA OPTION
      IF(ICLMAT.NE.1)GO TO 12
      RHA=RH
      UW=WNDVEL
      WD=WNDDIR
      ICAT=IPASCT
   12 CONTINUE
      IF(DLEN.EQ.0.0)DLEN=10.0
      IF(BRATE.EQ.0.0)BRATE=(1.0/14.3)
      QLENTH=XN*DLEN
      BREXP=BRATE
      QMUN=XN*FW
      EMUN=QMUN*YF*EFF/100.0
      WS=UW
      IF(XMIS(1).LE.0.0)GO TO 4
      XREFZ=100.0
      SIGZR=XMIS(1)
      ZDIFF=XMIS(2)
      YDIFF=XMIS(3)
      HM=XMIS(4)
      HK=XMIS(5)
      RC=XMIS(6)
      VS=XMIS(7)
      GO TO 5
    4 CALL PARMS(ICAT)
    5 CONTINUE
C*****REDEFINE WIND PROFILE EXPONENT IF READ IN POSITIVE
      IF(WPOWR.GE.0.0)WINDP=WPOWR
      IF(WS.LE.0.0)  WS=0.1
      U2=WS
C-----------------------------------------------------------------------
C                          LOCATE GENERATING LINE
C-----------------------------------------------------------------------
      THETA=(HEAD-WD)*PIRAD
C*** LOCATE CENTER OF GENERATING LINE.
      YHEAD=(XNORTH+90.0-HEAD)*PIRAD
      XBURST=RNG*SIN(YHEAD)+XM
      YBURST=RNG*COS(YHEAD)+YM
      ZBURST=ZM
C-----------------------------------------------------------------------
C                            TRANSFORM TO OBSERVER COORDINATES
C-----------------------------------------------------------------------
C*** TRANSFORM TO ORIGIN UNDER OBSERVER AND X-AXIS UNDER TARGET.
      ANGLR=ATAN2(YT-YO,XT-XO)
      XI(1)=(XBURST-XO)*COS(ANGLR)+(YBURST-YO)*SIN(ANGLR)
      YI(1)=(YBURST-YO)*COS(ANGLR)-(XBURST-XO)*SIN(ANGLR)
      ZI(1) = ZBURST
      TTI(1) = TBURST
C*** GET THE WIND DIRECTION ANGLE WITH THE NEW X-AXIS.
      ANGLD = ANGLR*180./PI
      WX=-(WD+XNORTH+ANGLD)
      WX=AMOD(WX, 360.0)
      WDA = WX*PI/180.
C---------------------------------------------------------------------
C                         WRITE INPUT DATA AND HEADINGS
C---------------------------------------------------------------------
      WRITE(IOOUT,8001)
      WRITE(IOOUT,1000)
      WRITE(IOOUT,1001)WS,WD,ICAT,RHA,XNORTH
```

```
      WRITE(IOOUT,998)
      WRITE(IOOUT,995)
      WRITE(IOOUT,997)XO,EXTC(1),YO,EXTC(2),ZO,EXTC(3),
     *XT,EXTC(4),YT,EXTC(5),ZT,EXTC(6),EXTC(7)
      WRITE(IOOUT,1002)
      WRITE(IOOUT,1003)XM,SIGZR,YM,XREFZ,ZM,HM
      WRITE(IOOUT,1004)HEAD,HK,RNG,RC,XN,VS
      WRITE(IOOUT,1005)QMUN,WINDP,QLENTH,ZDIFF,BREXP,YDIFF
      WRITE(IOOUT,1006)EFF,YF
 8001 FORMAT(//,21X,36H*****INPUT*****ALL LENGTHS IN METERS,/,
     *21X,38(2H--))
 1000 FORMAT(21X,15HMETEOROLOGICAL:)
 1001 FORMAT(24X,10HWIND SPEED,10X,F6.1,1X,3HM/S,/,
     *          24X,14HWIND DIRECTION,6X,F6.1,1X,3HDEG,/,
     *          24X,17HPASQUILL CATEGORY,3X,I3,/,
     *          24X,17HRELATIVE HUMIDITY,3X,F6.1,1X,1H%,/,
     *          24X,21HNOTE: X AXIS HEADING:,1X,F6.1,1X,3HDEG,1X,
     *          28HCLOCKWISE FROM NORTH (DCWFN))
 1002 FORMAT(20X,19HTANK/MUNITION DATA:,16X,21HDIFFUSION PARAMETERS:)
 1003 FORMAT(24X,7HX(TANK),12X,F6.1,10X,10HSIGZ(XREF),21X,F6.1,/,
     *          24X,7HY(TANK),12X,F6.1,10X,4HXREF,27X,F6.1,/,
     *          24X,7HZ(TANK),12X,F6.1,10X,17HMIXING HEIGHT(HM),14X,F6.1)
 1004 FORMAT(24X,14HHEADING(DCWFN),5X,F6.1,10X,20HSCAVENGING COEFF(HK),
     *11X,F6.3,/,24X,5HRANGE,14X,F6.1,10X,20HREFLECTION COEFF(RC),11X,
     *F6.3,/,24X,10HNO GRNADES,9X,F6.1,10X,21HSETTLING VELOCITY(VS),
     *5X,F6.3,1X,4HCM/S)
 1005 FORMAT(24X,14HSMOKE MASS(GM),5X,F6.1,10X,29HVERTICAL WIND EXPONENT
     *(WPOWR),2X,F6.3,/,24X,11HLINE LENGTH,8X,F6.1,10X,
     *29HVERTICAL DIFF CONSTANT(ZDIFF),2X,F6.3,/,
     *24X,13HBURN CONSTANT,2X,F6.3,1X,3H1/S,10X,
     *30HCROSSWIND DIFF CONSTANT(YDIFF),1X,F6.3)
 1006 FORMAT(24X,10HEFFICIENCY,9X,F6.1,10X,12HYIELD FACTOR,17X,F6.1)
  998 FORMAT(21X,16HOBSERVER/TARGET:,8X,24HEXTINCTION COEFFICIENTS:)
  995 FORMAT(49X,7HMICRONS,3X,7HM**2/GM)
  997 FORMAT(24X,6HX(OBS),3X,F6.1,10X,7H0.4-0.7,3X,F6.3,/,
     *          24X,6HY(OBS),3X,F6.1,10X,7H0.7-1.2,3X,F6.3,/,
     *          24X,6HZ(OBS),3X,F6.1,10X,7H 1.06  ,3X,F6.3,/,
     *          24X,6HX(TAR),3X,F6.1,10X,7H3.0-5.0,3X,F6.3,/,
     *          24X,6HY(TAR),3X,F6.1,10X,7H8.0-12.,3X,F6.3,/,
     *          24X,6HZ(TAR),3X,F6.1,10X,7H 10.6  ,3X,F6.3,/,
     *          49X,4H94.0,1X,3HGHZ,2X,F6.3)
      WRITE(IOOUT,996)
  996 FORMAT(1H1,21X,16H*****OUTPUT*****,/,21X,38(2H--))
      WRITE(IOOUT,3000)
 3000 FORMAT(24X,4HTIME,6X,2HCL,23X,12HTRANSMISSION,/,24X,5H(SEC),2X,
     *9H(GM/M**2),2X,7H0.4-0.7,1X,7H0.7-1.2,3X,4H1.06,2X,7H3.0-5.0,1X,
     *7H8.0-12.,2X,4H10.6,4X,5H94GHZ)
C-----------------------------------------------------------------------
C                          BEGIN CL CALCULATIONS
C-----------------------------------------------------------------------
      DO 400 IT=ISTO,IETO,IDTO
      ITT=IT
C     * SET UP LOOP ON SPACIAL DISTRIBUTION
      XC = 0.0
      YC = 0.0
      ZC = ZO
      DELX=SQRT( (XT-XO)**2+(YT-YO)**2 )/IXMAX
      DELZ = (ZT-ZO)/IXMAX
C** FOR EACH TIME GET THE CONCENTRATION AT IXMAX POINTS ALONG LINE-OF-SI
            DO 300 IX=1,IXMAX
            XC = XC + DELX
            ZC = ZC + DELZ
            XA(1) = XC
            XA(2) = YC
            XA(3) = ZC
            T = FLOAT(IT)
            UBXB=U2
            CALL CONCN(XA,T,C)
            XDTA(IX)=XC
            CDTA(IX)=C
```

```
  300 CONTINUE
C*** INTEGRATE ALONG LINE-OF-SIGHT TO GET CL.
      CALL SUMA (XDTA,CDTA, CL ,IXMAX)
C-----------------------------------------------------------------------
C                                         CALCULATE TRANSMITTANCE
C-----------------------------------------------------------------------
      DO 500 J=1,7
      XTRAN(J)=EXP(-1.0*EXTC(J)*CL(IXMAX))
  500 CONTINUE
      WRITE(IOOUT,2000)T,CL(IXMAX),(XTRAN(J),J=1,7)
 2000 FORMAT(23X,F5.1,2X,F8.3,2X,7(1X,F7.3))
  400 CONTINUE
C*****
C*****SET TRANSMISSION FOR RETURN TO EOSAEL
C*****
      IF(KWAVE.GT.0)GO TO 3
      TRANS=1.0
      IFLAG=1
    3 TRANS=XTRAN(KWAVE)
      GO TO 6
 9999 WRITE(IOOUT,9000)
 9000 FORMAT(21X,28H*****PROGRAM GRNADE END*****)
      RETURN
      END
```

```
      SUBROUTINE CONCN(XA,T,C2)
C/****************************************************************/
C/*                         SUBROUTINE CONCN                     */
C/*                           GRNAD MODULE                       */
C/*                            EOSAEL80                          */
C/****************************************************************/
C/* PURPOSE:                                                     */
C/*       CALCULATES CONCENTRATION AT A SPECIFIED POSITON AND TIME. */
C/* USAGE:                                                       */
C/*       THE CONCENTRATION IS USED WITH THE EXTINCTION COEFFICIENT */
C/*       TO COMPUTE TRANSMITTANCE.                              */
C/* DESCRIPTION OF PARAMETERS:                                   */
C/*       XA          - POSITION IN METERS, INPUT.               */
C/*       T           - TIME IN SECONDS, INPUT.                  */
C/*       C2          - CONCENTRATION, OUTPUT.                   */
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:               */
C/*                   LOCAT               UMEAN                  */
C/* COMMON BLOCK STATEMENTS REQUIRED:                            */
C/*       MECH1       MECH2       MECH3                          */
C/* REMARKS:                                                     */
C/*       CONCN COMPUTES FIVE TERMS AND MULTIPLIES THEM TO       */
C/*       GET CONCENTRATION.                                     */
C/****************************************************************/
      DIMENSION XA(3)
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK
      COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
      COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH
      COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM
C-----------------------------------------------------------------------
C         CONCENTRATION=TERM1*TERM2*TERM3*TERM4*TERM5
C-----------------------------------------------------------------------
      C2 = 0.0
      X = XA(1)
      Y = XA(2)
      Z = XA(3)
C*** GET CROSSWIND AND DOWNWIND COMPONENTS OF GENERATING LINE.
      QC=ABS(QLENTH*COS(THETA))
      QD=ABS(QLENTH*SIN(THETA))
      NBPT=1
      DO 390 J = 1,NBPT
      TJ = T- TTI(J)
C-----------------------------------------------------------------------
C              TERM1 IS CLOUD MASS AS A FUNCTION OF TIME.
C-----------------------------------------------------------------------
      IF(BREXP*TJ.GE.200.0)STOP01
      TRM1=EMUN*(1.0-EXP(-BREXP*TJ))
C-----------------------------------------------------------------------
C                   TERM2 IS TERRAIN SCAVENGING TERM.
C-----------------------------------------------------------------------
C** CHANGE TO SMOKE COORDINATES.
      CALL LOCAT(J,X,Y,Z,XB,YB,ZB)
C*** UPWIND END OF GENERATING LINE WILL BE ORIGIN OF SMOKE SYSTEM.
      XB=XB+0.5*QD
      CALL UMEAN(J,TJ)
      IF(HK*XB/UBXB.GE.200.0)STOP01
      TRM2 = EXP(-HK*XB/UBXB)
C-----------------------------------------------------------------------
C                    TERM3 IS DOWNWIND PROBABILITY DENSITY.
C-----------------------------------------------------------------------
      UT=UBXB*TJ
      IF(XB.LE.0.) GO TO 999
      IF(XB.GT.0..AND.XB.LE.UT)  TERM3=1./(UT+0.5*QD)
      IF(XB.GT.UT.AND.XB.LE.UT+QD) TERM3=(UT+QD-XB)/(UT*QD+0.5*QD*QD)
      IF(XB.GT.UT+QD)  TERM3=0.
      TRM3=TERM3
C-----------------------------------------------------------------------
C                    TERM4 IS CROSSWIND PROBABILITY DENSITY.
C-----------------------------------------------------------------------
      YWIDTH=YDIFF*XB+QC
      TRM4= 1./YWIDTH
```

```
      IF(YB .GT. 0.5*YWIDTH )  TRM4=0.
      IF( YB .LT. -0.5*YWIDTH)  TRM4=0.
C-----------------------------------------------------------------------
C                    TERM5 IS VERTICAL PROBABILITY DENSITY.
C-----------------------------------------------------------------------
      SIGZT=SIGZR*(XB/XREFZ)**ZDIFF
      SZT2=SIGZT*SIGZT
      HMVT = ZI(J) - (VS/100.0)*(XB/UBXB)
      TV1 = EXP(-(HMVT-ZB)**2/(2.0*SZT2))
      TV2 = RC*EXP(-(HMVT+ZB)**2/(2.0*SZT2))
      TERR = 1.0E-06
      JC = 0
      T1 = 0.0
      T2 = 0.0
      T3 = 0.0
      T4= 0.0
      TS = 0.0
  370 JC = JC+1
      R1 = RC**(JC-1)
      R2 = RC**JC
      R3 = RC**(JC+1)
      IF( (2.*JC*HM+HMVT+ZB)**2 .GT. 600.*2.*SZT2)  GO TO 371
      T1 = R1*EXP(-(2.0*JC*HM-HMVT-ZB)**2/(2.0*SZT2))
      T2 = R2*EXP(-(2.0*JC*HM-HMVT+ZB)**2/(2.0*SZT2))
      T3 = R2*EXP(-(2.0*JC*HM+HMVT-ZB)**2/(2.0*SZT2))
      T4 = R3*EXP(-(2.0*JC*HM+HMVT+ZB)**2/(2.0*SZT2))
      T1234 = T1 + T2 + T3 + T4
      TS = TS + T1234
      IF(T1234 - TERR) 371,371,370
  371 CONTINUE
      TV3 = TS
      TRM5 = TV1+TV2+TV3
      TRM5=TRM5/( SIGZT*SQRT(2.*PI) )
      C1 = TRM1*TRM2*TRM3*TRM4*TRM5
      C2 = C2 + C1
  380 CONTINUE
  999 RETURN
      END
```

```
      SUBROUTINE GOGET(WAVE1,KWAVE)
C/**********************************************************************/
C/*                        SUBROUTINE GOGET                            */
C/*                          GRNAD MODULE                              */
C/*                            EOSAEL80                                */
C/**********************************************************************/
C*****SUBROUTINE FINDS SPECTRAL BAND FOR GIVEN SINGLE WAVELENGTH
      KWAVE=0
      IF(WAVE1.GE.0.40.AND.WAVE1.LT.0.70)KWAVE=1
      IF(WAVE1.GE.0.70.AND.WAVE1.LT.1.20)KWAVE=2
      IF(WAVE1.GE.3.00.AND.WAVE1.LT.5.00)KWAVE=4
      IF(WAVE1.GE.8.00.AND.WAVE1.LT.12.0)KWAVE=5
      IF(WAVE1.EQ.1.06)                  KWAVE=3
      IF(WAVE1.EQ.10.6)                  KWAVE=6
      IF(WAVE1.EQ.94.0)                  KWAVE=7
      IF(WAVE1.GT.3188.0.AND.WAVE1.LT.3195.0)KWAVE=7
      RETURN
      END
```

```
      SUBROUTINE PARMS(ICAT)                                            PAR00010
C/*******************************************************************/PAR00020
C/*                    SUBROUTINE PARMS                              */PAR00030
C/*                      GRNAD MODULE                                */PAR00040
C/*                        EOSAEL80                                  */PAR00050
C/*******************************************************************/PAR00060
C*****SETS DIFFUSION PARAMETER DEFAULTS AS FUNCTION OF PASQUILL CATEGORYPAR00070
      COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM            PAR00080
      YDIFF=0.355                                                       PAR00090
      HK=0.002                                                          PAR00100
      VS=0.021                                                          PAR00110
      RC=0.70                                                           PAR00120
      XREFZ=100.0                                                       PAR00130
      IF(ICAT.GT.3)GO TO 1                                              PAR00140
      WINDP=0.10                                                        PAR00150
      ZDIFF=2.08                                                        PAR00160
      HM=1000.0                                                         PAR00170
      SIGZR=14.0                                                        PAR00180
      GO TO 3                                                           PAR00190
    1 IF(ICAT.GT.4)GO TO 2                                              PAR00200
      WINDP=0.20                                                        PAR00210
      ZDIFF=1.40                                                        PAR00220
      HM=300.0                                                          PAR00230
      SIGZR=7.2                                                         PAR00240
      GO TO 3                                                           PAR00250
    2 WINDP=0.40                                                        PAR00260
      ZDIFF=1.04                                                        PAR00270
      HM=50.0                                                           PAR00280
      SIGZR=5.0                                                         PAR00290
    3 RETURN                                                            PAR00300
      END                                                               PAR00310
```

```
      SUBROUTINE EXTIN(EX)
C/*****************************************************************/
C/*                        SUBROUTINE EXTIN                        */
C/*                          GRNAD MODULE                          */
C/*                            EOSAEL80                            */
C/*****************************************************************/
C*****PROGRAM TO SET EXTINCTION COEFFICIENTS FOR WP/RP SMOKE****
      DIMENSION EX(7),CX(7)
      DATA CX/4.304,2.166,1.541,0.350,0.338,0.364,0.001/
      DO 1 I=1,7
      EX(I)=CX(I)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE UMEAN(J,TJ)                                            UME00010
C/*******************************************************************/UME00020
C/*                       SUBROUTINE UMEAN                           */UME00030
C/*                         GRNAD MODULE                             */UME00040
C/*                           EOSAEL80                               */UME00050
C/*******************************************************************/UME00060
C*****CALCULATES MEAN WIND SPEED OVER EXTENT OF CLOUD                   UME00070
      COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP                  UME00080
      COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH                             UME00090
      COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM            UME00100
      IC = 0                                                            UME00110
      QD = ABS(QLENTH*SIN(THETA))                                       UME00120
      P = WINDP + 1.0                                                   UME00130
      C1=P*2.0**WINDP                                                   UME00140
      Z2=ZI(J)                                                          UME00150
      IF(Z2-2.0)10,10,20                                                UME00160
   10 CONTINUE                                                          UME00170
      Z2=2.0                                                            UME00180
      Z1=0.01                                                           UME00190
      GO TO 200                                                         UME00200
   20 Z1=ZI(J)-1.5*SIGZR                                                UME00210
      IF(Z1)30,30,40                                                    UME00220
   30 Z1=0.01                                                           UME00230
   40 CONTINUE                                                          UME00240
      UH = (U2/((Z2-Z1)*C1))*(Z2**P-Z1**P)                              UME00250
      US=UH                                                             UME00260
   50 CONTINUE                                                          UME00270
      XC = UH*TJ                                                        UME00280
      Z2 = ZI(J) - (VS/100.0)*(XC/UH)                                   UME00290
      IF(Z2-2.0)151,151,155                                             UME00300
  151 CONTINUE                                                          UME00310
      Z2=2.0                                                            UME00320
      Z1=0.01                                                           UME00330
      GO TO 160                                                         UME00340
  155 CONTINUE                                                          UME00350
      SIGZT = SIGZR*((XC+QD)/XREFZ)**ZDIFF                              UME00360
      Z1 = ZI(J) - 1.5*SIGZT                                            UME00370
      IF(Z1)159,159,160                                                 UME00380
  159 CONTINUE                                                          UME00390
      Z1=0.01                                                           UME00400
  160 UBXJ = (U2/((Z2-Z1)*C1))*(Z2**P-Z1**P)                            UME00410
      GO TO 1000                                                        UME00420
  200 CONTINUE                                                          UME00430
      US=U2                                                             UME00440
      UH=US                                                             UME00450
      UBXJ=US                                                           UME00460
 1000 CONTINUE                                                          UME00470
      UBXJ = SQRT((UBXJ**2 + UH**2)/2.0)                                UME00480
      IC = IC + 1                                                       UME00490
      IF(IC .EQ.1) UH = UBXJ                                            UME00500
      IF(IC.EQ.1) GO TO 50                                              UME00510
      UBXB=UBXJ                                                         UME00520
      RETURN                                                            UME00530
      END                                                               UME00540
```

```
      SUBROUTINE SUMA(X,Y,Z,NDIM)
C/*********************************************************************/
C/*                           SUBROUTINE SUMA                        */
C/*                             GRNAD MODULE                         */
C/*                               EOSAEL80                           */
C/*********************************************************************/
C/*                                                                  */
C/* PURPOSE:                                                         */
C/*      GENERAL PURPOSE INTEGRATION SUBROUTINE                      */
C/* USAGE:                                                           */
C/*      CALLED FROM GRNAD TO GET INTEGRAL OF CONCENTRATION ALONG    */
C/*      LINE-OF-SIGHT.                                              */
C/* DESCRIPTION OF PARAMETERS:                                       */
C/*      X          INDEPENDENT VARIABLE                             */
C/*      Y          DEPENDENT VARIABLE                               */
C/*      Z          INTEGRAL OF Y OVER X                             */
C/*      INDIM      NUMBER OF POINTS                                 */
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:                   */
C/*      NONE                                                        */
C/* COMMON BLOCK STATEMENTS REQUIRED:                                */
C/*      NONE                                                        */
C/* REMARKS:                                                         */
C/*      AS USED BY GRENADE, X IS POSITION ONLINE-OF-SIGHT, Y IS     */
C/*      LOCAL CONCENTRATION, Z IS TOTAL CONCENTRATION.              */
C/* METHOD:                                                          */
C/*      SUMA INCREMENTS INTEGRAL BY AVERAGE OF LAST TWO Y VALUES    */
C/*      TIMES DELTA X.                                              */
C/*********************************************************************/
      DIMENSION X(1000),Y(1000),Z(1000)
      SUM2          = 0.0
      IF              (NDIM - 1) 4,3,1
C                     *** INTEGRATION LOOP ***
1     DO 2 I        = 2,NDIM
        SUM1        =   SUM2
        SUM2        =   SUM2 + 0.500*(X(I)-X(I-1))*(Y(I)+Y(I-1))
2       Z(I-1)      =   SUM1
3     Z(NDIM)       = SUM2
4     RETURN
      END
```

```
      SUBROUTINE LOCAT(I,X,Y,Z,XB,YB,ZB)
C/*********************************************************************/
C/*                            SUBROUTINE LOCAT                      */
C/*                              GRNAD MODULE                        */
C/*                                EOSAEL80                          */
C/*********************************************************************/
C/* PURPOSE:                                                         */
C/*      LOCAT TRANSLATES POSITION INTO BURST AND WIND COORDINATES.  */
C/* USAGE:                                                           */
C/*      CALLED BY CONCN                                             */
C/* DESCRIPTION OF PARAMETERS:                                       */
C/*      T            TIME (SECONDS), INPUT                          */
C/*      X,Y,Z        ORIGINAL POSITION (METERS), INPUT              */
C/*      XI,YI,ZI     ORIGINAL BURST POSITION (METERS), INPUT        */
C/*      WDA          ANGLE BETWEEN WIND VECTOR AND LINE OF SIGHT    */
C/*                   (RADIANS), INPUT                               */
C/*      XB,YB,ZB     TRANSLATED POSITION (METERS), OUTPUT           */
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:                   */
C/*      NONE                                                        */
C/* COMMON BLOCK STATEMENTS REQUIRED:                                */
C/*      MECH1        MECH2                                          */
C/* REMARKS:                                                         */
C/*      ORIGINAL COORDINATES HAVE ORIGIN AT OBSERVER AND X-AXIS     */
C/*      THROUGH TARGET.  NEW COORDINATES HAVE ORIGIN AT BURST       */
C/*      AND X-AXIS IN DIRECTION OF WIND VECTOR.                     */
C/* METHOD:                                                          */
C/*      STANDARD ROTATION AND TRANSLATION OF AXES                   */
C/*********************************************************************/
      COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
      COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH
      XB = (X -XI(I))*COS(WDA) + (Y-YI(I))*SIN(WDA)
      YB =-(X-XI(I))*SIN(WDA) + (Y - YI(I))*COS(WDA)
      ZB=Z
      IF(ZB.LT.0.0.AND.ABS(ZB).GT.ZI(I)) ZB=0.0
      RETURN
      END
```

```
      SUBROUTINE DATRD(IWRIT,IFLAG)
C/*********************************************************************/
C/*                       SUBROUTINE DATRD                            */
C/*                         GRNAD MODULE                              */
C/*                           EOSAEL80                               /*/
C/*********************************************************************/
C*****THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS
C     THE SMOKE(EOSAEL) PROGRAM
C     INPUTS
C        EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,
C        FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.
C        PER FIELD BEGINNING IN COL 11.  THE CARDS ARE NOT ORDER
C        DEPENDENT.
C        NAME                  IGNORED
C        BURN                  IGNORED
C        MUNC      XM,YM,ZM    COORDINATES OF GRENADE FIRING TANK
C                  HEAD        HEADING OF GRENADE FIRING TANK CLOCKWISE
C                              FROM NORTH
C                  RNG         GRENADE FIRING RANGE OF TANK
C                  DLEN        SPACING OF GRENADES ALONG LINE PERPENDICULAR
C                              TO HEADING
C        OBSC      XO,YO,ZO    COORDINATES OF THE OBSERVER   (M,M,M)
C        TARC      XT,YT,ZT    COORDINATES OF THE TARGET    (M,M,M)
C        BART
C                  STO         STARTING TIME  (ELAPSED TIME SINCE BLAST)
C                  ETO         ENDING TIME FOR CALCULATION
C                  DTO         TIME INCREMENT FOR CALCULATION
C                  XNORTH      X AXIS HEADING CLOCKWISE FROM NORTH
C        MUNT
C                  XN          NUMBER OF MUNITIONS FIRED AT THE SAME
C                              LOCATION AND AT THE SAME TIME
C                  FW          FILL WEIGHT )LBS)
C                  TBURN       BURN TIME OF SMOKE TYPE (IGNORED)
C                  ITYPE       TYPE OF SMOKE (DEFAULTS TO 1)
C                              1.=WP, 2.=PWP, 3.=HC, 4.=FOG OIL
C                  EFF         EFFICIENCY OF BURN (PERCENT).  IF 0.0,
C                              DEFAULTS TO 62.0%.
C                  YF          YIELD FACTOR IF 0.0,DEFAULTS TO ANALYTICAL
C                              MODEL
C                  BRATE       EXPONENTIAL BURN RATE PARAMETER
C        METR
C                  RHA         RELATIVE HUMIDITY    (PERCENT)
C                  UW          WIND VELOCITY     (M/S)
C                  WD          WIND DIRECTION  (DEGREES)
C                  ICAT        PASQUILL CATEGORY
C                                 1-A, 2-B, 3-C, 4-D, 5-E, 6-F
C                  AIRT        SURFACE AIR TEMPERATURE (IGNORED)
C                  TGRAD       VERT TEMP GRADIENT (IGNORED)
C                  WPOWR       WIND PROFILE EXPONENT (DIMENSIONLESS)
C        EXTC                  DESIRED CHANGES IN EXTINCTION COEFF.
C                              (OPTIONAL). IF NOT USED OR READ AS 0.
C                              DEFAULTS TO ALPHA ARRAY VALUE IN STRNS.
C                               BANDS ARE:
C                                  0.4-0.7 MICRONS
C                                  0.7-1.2 MICRONS
C                                     1.06  MICRONS
C                                  3.0-5.0 MICRONS
C                                  8.0-12. MICRONS
C                                     10.6  MICRONS
C                                     94.0  GHZ.
C        MISC                  DIFFUSION PARAMETER OPTION CARD FOR GRNAD
C                  SIGZR       DOWNWIND REFERENCE AT 100 M REFERENCE DIST.
C                  ZDIFF        VERTICAL DIFFUSION COEFFICIENT
C                  YDIFF       CROSSWIND DIFFUSION COEFFICIENT
C                  HM          HEIGHT OF MIXING LAYER  (METERS)
C                  HK          TERRAIN SCAVENGING COEFFICIENT
C                  RC          TERRAIN REFLECTION COEFFICIENT
C                  VS          SETTLING VELOCITY   (CM SEC)
C        GO            SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT
C        DONE          END OF JOB.
```

```
C*************************************************************************
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /GEOMET/PTS(15),IGEOSW
      COMMON/CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,
     *              WNDVEL,WNDDIR,IPASCT
      COMMON/MECHO/XM,YM,ZM,XO,YO,ZO,XT,YT,ZT,ISTO,IETO,IDTO,XN,FW,
     *TBURN,ITYPE,EFF,YF,RHA,UW,WD,ICAT,AIRT,TGRAD,BRATE,HEAD,RNG,
     *DLEN,WPOWR,EXTC(8),XMIS(8),XNORTH
      DIMENSION DUMY(8),IR(26),IR1(10),R1(10),EX(7),INAME(35)
      DATA IR/2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA,
     *2HRC,2HEX,2HTC,2HBU,2HRN,2HMI,2HSC,2HGO,2H  ,2HDO,2HNE,2HNA,2HME,
     *2HOU,2HTP/
      IF (IFLAG.GT.0) GO TO 8
      DO 2 J=1,8
      XMIS(J)=0.0
    2 EXTC(J)=0.
C***
C***  BEGINNING OF READ LOOP
C***
8     CONTINUE
      IF(IWRIT.EQ.0)GO TO 6
      WRITE(IOOUT,200)
  200 FORMAT(1H0,21X,20H*****CARD INPUT*****,/,21X,40(2H--))
  201 FORMAT(2A2,6X,35A2)
  202 FORMAT(1H0,21X,2A2,6X,35A2)
    6 DO 70 I=1,13
      IF(I.EQ.13) GO TO 310
      IF(IFLAG.GT.0) GO TO 4
      IFLAG=1
      READ(IOIN,201)IR1(1),IR1(2),(INAME(J),J=1,35)
      IF(IWRIT.EQ.0) GO TO 4
      WRITE(IOOUT,202)IR1(1),IR1(2),(INAME(J),J=1,35)
    4 READ(IOIN,20)IR1(1),IR1(2),(R1(J),J=2,8)
      IF(IWRIT.EQ.0) GO TO 5
      WRITE(IOOUT,30)IR1(1),IR1(2),(R1(J),J=2,8)
    5 IF(IR1(1).EQ.IR(21).AND.IR1(2).EQ.IR(22)) GO TO 998
   20 FORMAT(2A2,6X,7F10.3)
   30 FORMAT(1H0,21X,2A2,6X,7F10.3)
C*****
C***  RELATING INPUT DATA TO VARIABLE NAMES.
C*****
              IF(IR1(1).EQ.IR(1).AND.IR1(2).EQ.IR(2))   GO TO 90
              IF(IR1(1).EQ.IR(3).AND.IR1(2).EQ.IR(4))   GO TO 100
              IF(IR1(1).EQ.IR(5).AND.IR1(2).EQ.IR(6))   GO TO 110
              IF(IR1(1).EQ.IR(7).AND.IR1(2).EQ.IR(8))   GO TO 120
              IF(IR1(1).EQ.IR(9).AND.IR1(2).EQ.IR(10))  GO TO 130
              IF(IR1(1).EQ.IR(11).AND.IR1(2).EQ.IR(12)) GO TO 140
              IF(IR1(1).EQ.IR(13).AND.IR1(2).EQ.IR(14)) GO TO 150
              IF(IR1(1).EQ.IR(15).AND.IR1(2).EQ.IR(16)) GO TO 155
              IF(IR1(1).EQ.IR(17).AND.IR1(2).EQ.IR(18)) GO TO 165
              IF(IR1(1).EQ.IR(19).AND.IR1(2).EQ.IR(20)) GO TO 175
              IF(IR1(1).EQ.IR(21).AND.IR1(2).EQ.IR(22)) GO TO 998
              IF(IR1(1).EQ.IR(23).AND.IR1(2).EQ.IR(24)) GO TO 70
              IF(IR1(1).EQ.IR(25).AND.IR1(2).EQ.IR(26)) GO TO 70
C*****
C     ERROR CAUTION FOR INVALID DATA CARD
C*****
      IFLAG=2
      WRITE(IOOUT,80)
   80 FORMAT(21X,35H*****CAUTION***** INVALID DATA CARD)
      GO TO 70
   90         RHA   = R1(2)
              UW    = R1(3)
              WD    = R1(4)
              ICAT  =IFIX(R1(5))
              AIRT  = R1(6)
              TGRAD = R1(7)
              WPOWR=R1(8)
      GO TO 70
  100         XN    = R1(2)
```

```
      FW     =453.6*R1(3)
      TBURN = R1(4)
      ITYPE = IFIX(R1(5))
      EFF   = R1(6)
      YF=R1(7)
      BRATE=R1(8)
      GO TO 70
 110  ISTO  = IFIX(R1(2))
      IETO  = IFIX(R1(3))
      IDTO  = IFIX(R1(4))
      XNORTH= R1(5)
      IF(ISTO.LE.0)ISTO=1
      GO TO 70
 120  XM    = R1(2)
      YM    = R1(3)
      ZM    = R1(4)
      HEAD  = R1(5)
      RNG   = R1(6)
      DLEN  = R1(7)
      GO TO 70
 130  XO    = R1(2)
      YO    = R1(3)
      ZO    = R1(4)
      GO TO 70
 140  XT    = R1(2)
      YT    = R1(3)
      ZT    = R1(4)
C*****
C*****BURN CARD DATA DUMMYED BY PROGRAM GRNAD*****
C*****
 155  DO 156 J=1,7
 156  DUMY(J)=R1(J+1)
      GO TO 70
 150  DO 152 J=1,7
 152  EXTC(J)=R1(J+1)
      GO TO 70
 165  DO 166 J=1,7
 166  XMIS(J)=R1(J+1)
  70  CONTINUE
 175  GO TO 311
C*****
C*****CAUTION FOR TOO MANY CARDS
C*****
 310  WRITE(IOOUT,320)
      IFLAG=3
C*****
C*****DEFAULT NON USER DEFINED INPUT*****
C*****
 311  IF(ITYPE.EQ.1)GO TO 3
      ITYPE=1
      WRITE(IOOUT,171)
 171  FORMAT(1H ,21X,17H*****CAUTION*****,/,1H ,21X,54HWRONG SMOKE TYPE
     *FOR PROGRAM GRNAD--DEFAULTED TO WP/RP)
   3  IF(EFF.EQ.0.0)EFF=62.0
      IF(YF.EQ.0.0)YF=3.14+0.032*RHA
      IF(EXTC(1).GT.0.0)GO TO 1
      CALL EXTIN(EX)
      DO 7 I=',7
   7  EXTC(I)=EX(I)
   1  CONTINUE
 320  FORMAT(21X,17H*****CAUTION*****,/,
     *21X,56HMORE THAN 10 DATA CARDS ENTERED--REMAINING CARDS IGNORED)
      GO TO 9999
 998  IFLAG=4
9999  IF(IGEOSW.NE.1) GO TO 555
      DISKTM=1000.
C***  CONVERT KM TO M.
      XT=PTS(1)*DISKTM
      YT=PTS(2)*DISKTM
      ZT=PTS(3)*DISKTM
```

```
      X0=PTS(4)*DISKTM
      Y0=PTS(5)*DISKTM
      Z0=PTS(6)*DISKTM
  555 RETURN
      END
```

```
      SUBROUTINE LT4M(H1,H2,ANGLE,ITYPE,IXY,TRAN,RADA,RADG,IEMISS,LEN,  L4M00010
     +MODEL,VIS,V11,V22,T1,ICLMAT,IERR,NR,IHAZE,MULDV)                  L4M00020
      LOGICAL ISPOT,LOREAD,N16                                          L4M00030
      COMMON /CONST/PI,PI2,CA,TWOPI,TORRMB,CDEGK                        L4M00040
      COMMON /M01/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL     L4M00050
      COMMON /M02/ WO(34),RO,TBOUND,JP,IM,ML,IP,JSTOR                   L4M00060
      COMMON /M05/ C1(501),C2(258),C3(86),C4(33),C5(6),C5DUM(9),C8(82), L4M00070
     1       C11(4),C12(15),C14(21),C15(6)                              L4M00080
      COMMON /M07/TR(67),FW(67),FO(67)                                  L4M00090
      COMMON /M08/SUM4,SUM5,SUM8,SUM11,SUM6                             L4M00100
      COMMON /M09/RADMAX,RADMIN,VRMAX,VRMIN                             L4M00110
      COMMON /M03/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),     L4M00120
     1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2                                 L4M00130
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUL4M00140
      COMMON /LOWEX/WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX,    L4M00150
     1 CLDHGT,NCLD                                                      L4M00160
      COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1                    L4M00170
      COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL                       L4M00180
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),     L4M00190
     1 BE(16),SINGWV,PF(65),LMAX                                        L4M00200
      COMMON /SPOTLO/ISPOT,LOREAD,N16                                   L4M00210
      DIMENSION TRAN(16)                                                L4M00220
      DIMENSION RADA(16),RADG(16)                                       L4M00230
C     PLANCK RADIANCE FUNCTION .........                                L4M00240
      FF(T,V)=1.190956E-16*(V**5)/(EXP(1.43879*V/T)-1.)                 L4M00250
C       WATT CM-2 ST-1 MICRON-1                                         L4M00260
C***********************************************************************L4M00280
C                                                                       L4M00290
C     PROGRAM MODIFIED LOWTRAN CALCULATES THE TRANSMITTANCE             L4M00300
C     OF THE ATMOSPHERE FROM 830 TO 1250, 2010 TO 3330, AND             L4M00310
C     5010 TO 39990 CM-1 (0.25 TO 2.0,3.0 TO 5.0,AND 8.0 TO 12.0        L4M00320
C     MICRONS) AT 20 CM-1 SPECTRAL INTERVALS ON A LINEAR WAVENUMBER     L4M00330
C     SCALE.                                                            L4M00340
C     REFRACTION AND EARTH CURVATURE EFFECTS ARE EXCLUDED. ATMOSPHERE   L4M00350
C     IS LAYERED IN ONE KM INTERVALS BETWEEN 0 AND 25 KM, 5 KM INTER-   L4M00360
C     VALS FROM 25 TO 50 KM, A 20 KM LAYER FROM 50 TO 70 KM, A 30 KM    L4M00370
C     LAYER FROM 70 TO 100 KM, AND ONE FROM 100 KM TO INFINITY.         L4M00380
C***********************************************************************L4M00390
C     PROGRAM ACTIVATED BY SUBMISSION OF CARD SEQUENCE AS FOLLOWS       L4M00400
C                                                                       L4M00410
C  CARD 1   MODEL,IHAZE,ITYPE,LEN,JP,NPLT,IM,ML,IEMISS,                 L4M00420
C           RO,TBOUND,BETAEX     FORMAT ( 9I3,3F10.3)                   L4M00430
C           MODEL    =0,METEOROLOGICAL DATA SPECIFIED                   L4M00440
C                    =1,TROPICAL MODEL ATMOSPHERE                       L4M00450
C                    =2,MIDLATITUDE SUMMER                              L4M00460
C                    =3,MIDLATITUDE WINTER                              L4M00470
C                    =4,SUBARCTIC SUMMER                                L4M00480
C                    =5,SUBARCTIC WINTER                                L4M00490
C                    =6,1962 US STANDARD                                L4M00500
C                    =7,NEW MODEL ATMOSPHERE                            L4M00510
C                    =8,ISRAELI STANDARD ATMOSPHERE (YEAR, DAYTIME)     L4M00520
C                    =9,ISRAELI STANDARD ATMOSPHERE (YEAR, NIGHTTIME)   L4M00530
C  ** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK **L4M00540
C     FOR SLANT PATHS IHAZE = 1,2, OR 3 ARE THE ONLY ALLOWED VALUES.    L4M00550
C              IHAZE    =0,NO AEROSOL ATTENUATION                       L4M00560
C                       =1, MARITIME POLAR                              L4M00570
C                       =2, MARITIME ARCTIC                             L4M00580
C                       =3, CONTINENTAL POLAR                           L4M00590
C                       =4, RAIN                                        L4M00600
C                       =5, SNOW                                        L4M00610
C                       =7,  USER SUPPLIED EXTINCTION COEFFICIENT       L4M00620
C                         (READ ON ATM CARD - SEE CARD 3 BELOW)         L4M00630
C           ITYPE    =1,HORIZONTAL (CONSTANT PRESSURE) PATH             L4M00640
C                    =2,VERTICAL OR SLANT PATH BETWEEN 2 ALTITUDES      L4M00650
C                    =3,VERTICAL OR SLANT PATH TO SPACE                 L4M00660
C           LEN      =0,NORMAL OPERATION                                L4M00670
C                    =1,DOWNWARD LONG PATH                              L4M00680
C           JP       =0,NORMAL OPERATION                                L4M00690
C                    =1,SUPPRESS PRINT OF HORIZ AND VERTICAL PROFILES   L4M00700
C          NPLT      =0, NORMAL OPERATION                               L4M00710
```

```
C                    =1, IN TRANSMISSION MODE WRITE, WAVELENGTH (UM),
C                        H2O, CO2+, OZONE, N2 C, H2O C, MOL SCAT,
C                        NITRIC, SO2, HNO3, NO2.  IN EMISSION MODE
C                        WRITE WAVELENGTH (UM) AND RADIANCE PER MICRON.
C             RESULTS WILL BE WRITTEN ON NPLOTU (SEE COMMON BLOCK IOUNIT)
C           IM       =1,RADIOSONDE DATA TO BE READ INITIALLY
C                    =0,NORMAL OPERATION OR WHEN SUBSEQUENT CALCULATIONS
C                   ARE TO BE RUN WITH MODEL = 7
C           ML       =NUMBER OF LEVELS TO BE READ IN FOR MODEL = 7
C   ***IM AND ML ONLY USED WHEN MODEL = 7  AND THEN ONLY ON
C      FIRST CALCULATIONS WHEN DATA READ IN
C           IEMISS  DETERMINES MODE OF EXECUTION OF PROGRAM
C                    =0,TRANSMITTANCE MODE
C                    =1,RADIANCE MODE
C           RO       RADIUS OF THE EARTH (KM) AT LOCATION OF CALCULATION
C    ***DEFAULT WILL BE MIDLATITUDE VALUE OF 6371.23 KM WHEN
C                     MODEL = 0 OR = 7   OTHERWISE DEFAULT IS EARTH RADIUS
C                     FOR STANDARD MODEL ATMOSPHERE SPECIFIED BY MODEL
C           TBOUND  TEMPERATURE OF EARTH (DEGREES K) AT LOCATION OF CALCUL
C    ***USED ONLY IN RADIANCE MODE FOR SLANT PATHS WHICH INTERSECT EARTH
C    ***DEFAULT IS TEMPERATURE OF FIRST LAYER   BOUNDARY TEMPERATURE
C         BETAEX  USER SUPPLIED EXTINCTION COEFFICIENT, INPUT ONLY
C                 WHEN IHAZE=7
C   CARD 2  H1,H2,ANGLE,RANGE,BETA,VIS,CLDHGT  FORMAT (7F10.3)
C           H1       INITIAL ALTITUDE (KM)
C           H2       FINAL ALTITUDE (KM)
C           ANGLE    INITIAL ZENITH ANGLE (DEG)
C           RANGE    PATH LENGTH (KM)
C           BETA     EARTH CENTER ANGLE SUBTENDED BY H1 AND H2 (DEG)
C           VIS      SEA LEVEL VISUAL RANGE (KM)
C           CLDHGT   HEIGHT OF BOTTOM OF CLOUD LAYER (KM), WHEN IHAZE NE 0
C    ***VIS NOT REQUIRED ON THIS CARD IF ICLMAT (EOMAIN) =1 OR
C    ***THIS IS FIRST LOOP THROUGH LT4
C    ***SEE MANUAL FOR MORE DETAIL
C   CARD 2A  V1,V2,MULDV       FORMAT (2F10.3,I2)
C           V1       INITIAL FREQUENCY (CM**-1)
C           V2       FINAL FREQUENCY (CM**-1)
C           MULDV    MULTIPLIER FOR FREQUENCY INCREMENT, WHERE THE
C                    INCREMENT IS A MULTIPLE OF 20 (CM**-1).
C   OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY NR=1 IN EOMAIN)
C      CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2).
C      CARDS 2 - NUMBER OF VALUES: FORMAT (2(E10.4,1X))
C         ONE VALUE OF WAVELENGTH (UM) AND RESPONSE FUNCTON PER CARD
C   CARD 3  IXY      FORMAT (I3)
C           IXY      =0,EXIT LOWTRAN MODULE
C                    =1,SELECT NEW WAVE FREQUENCY RANGE (CARD 2A)
C                    =2,SELECT NEW DATA SEQUENCE (CARDS 1,2,2A,3)
C                    =3,SELECT NEW CARD 2 AND CARD 3
C                    =4,SELECT NEW CARD 1 AND CARD 3
C
C   ***FOR NON-STANDARD CONDITIONS SEE MANUAL
C*******************************************************************
      V1=V11
      V2=V22
      KMAX=16
      ISPOT1=0
      RESPFN=0.
      SUMRPF=0.
      SUMINT=0.
      IF (ISPOT) NPLT=0
  200 CONTINUE
      IF (.NOT.LOREAD) GO TO 400
      LOREAD=.FALSE.
      READ (LOUNIT,3300) IATM,NL
      NL4=4*NL
      DO 299 I=1,NL4
  299 READ (LOUNIT,3500) DUMMY
      READ (LOUNIT,3510) PPMSO2,PPMNH3,PPMNO2
      READ (LOUNIT,3700)(TR(I),FW(I),FO(I),I=1,67)
      READ (LOUNIT,3800)(C1(I),I=1,501)
```

```
      READ (LOUNIT,3800)(C2(I),I=1,258)
      READ (LOUNIT,3800)(C3(I),I=1,86)
      READ (LOUNIT,3900)(C4(I),I=1,33)
      READ(LOUNIT,3810)(C5(I),I=1,6)
      READ (LOUNIT,3900)(C8(I),I=1,82)
      READ (LOUNIT,4000)(C11(I),I=1,4)
      READ (LOUNIT,4010)(C12(I),I=1,15)
      READ (LOUNIT,4020)(C14(I),I=1,21)
      READ (LOUNIT,4020)(C15(I),I=1,6)
      READ (LOUNIT,4010)(FS(I),S1(I),S2(I),I=1,9)
      READ (LOUNIT,4010)(FNH3(I),FH1(I),FH2(I),I=1,9)
      READ (LOUNIT,4010)(FNO2(I),O1(I),O2(I),I=1,9)
      REWIND LOUNIT
      IF (ISPOT) GO TO 710
400   CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN(1))
      IF (IERR.EQ.1) RETURN
      JP=0
      JSTOR=0
      IP=0
      IF (.NOT.ISPOT) GO TO 700
      BETA=0.
      RANGE=0.
      RO=0.
      IF (ITYPE.EQ.1) RANGE=ANGLE
      IF (IXY.EQ.0) GO TO 700
401   GO TO (500,700,600,680),IXY
500   AVW=1.E+04/V1
      ALAM=1.E+04/V2
      SUMA=0.
      GO TO 1100
  600 IF (MODEL.EQ.0) GO TO 800
      ISPOT1=1
      GO TO 1000
  700 CONTINUE
      RE=6371.23
680   IF (.NOT.ISPOT) READ (IOIN,3300) MODEL,IHAZE,ITYPE,
     1    LEN,JP,NPLT,IM,ML,IEMISS,RO,TBOUND
C*****IEMISS=0=TRANSMISSION MODE / IEMISS=1=EMISSION MODE
      IF ((IEMISS.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,4100)
      IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,4200)
      IF (ISPOT) GO TO 800
      IF(MODEL.EQ.0.OR.MODEL.EQ.7) GO TO 210
  710 READ(LOUNIT,3300)IATM,NL
      MSKIP=(MODEL-1)/2
      IF(MSKIP.EQ.0) GO TO 220
      IF (MSKIP.EQ.4) MSKIP=3
      DO 230 J=1,MSKIP
      DO 230 I=1,NL
  230 READ(LOUNIT,3500) DUMMY
  220 CONTINUE
C     ISRAELI STD ATM READS
      IF (MODEL.EQ.8) GO TO 270
      IF (MODEL.EQ.9) GO TO 250
      IF (2*(MODEL/2).EQ.MODEL) GO TO 250
  270 DO 240 I=1,NL
  240 READ(LOUNIT,3500)Z(I),P(I),T(I),WA(I),WH(I),WO(I)
      GO TO 210
  250 DO 260 I=1,NL
  260 READ(LOUNIT,3550)Z(I),P(I),T(I),WA(I),WH(I),WO(I)
  210 REWIND LOUNIT
      IF (ISPOT) RETURN
  800 M=MODEL
  900 IF (RO.GT.0) RE=RO
      LENTOR=LEN
1000  CALL ABSORB(IXY,IERR,W,V1,V2,DV,SUMA,MULDV,ANGLE,LEN,ITYPE,H1,H2,
     1      MODEL,ISPOT1,RANGE,BETA,VIS,ICLMAT,IV1,IV2,IDV)
      IF (IERR.EQ.1) TRAN(1)=1.
      IF (IERR.EQ.1) RETURN
 1100 CONTINUE
      IF (.NOT.ISPOT) WRITE (IOOUT,4300)
```

```
      IF (.NOT.ISPOT) WRITE(IOOUT,4400) (W(I),I=1,6),W(8),W(10)
      IF (.NOT.ISPOT) WRITE(IOOUT,4401)
      IF (.NOT.ISPOT) WRITE (IOOUT,4500) W(11),W(12),W(13),W(14),
     * W(15),W(16)
 1200 CONTINUE
C     NCLD WILL BE THE INDEX OF THE LAYER ABOVE CLDHGT
      DO 5 ICLD=2,6
      NCLD=ICLD
      IF (CLDHGT.LT.Z(ICLD)) GO TO 6
5     CONTINUE
6     CONTINUE
      IF (CLDHGT.GT.Z(6)) WRITE (IOOUT,7)
7     FORMAT (1H ,25H**** WARNING FROM LOWTRAN,/,1X,14HCLOUD BASE IS ,
     1  23HLIMITED TO 4 KM MAXIMUM/)
      I=1
      L=1
      IV=IV1
      ICOUNT=0
      IF (N16) KWAVE=0
      IF (IEMISS.EQ.0) GO TO 1300
      RADSUM=0.0
      FACTOR=0.5
      CALL LTPATH(WLAY,WPATH,TBBY,ANGLE,LEN,ITYPE,H1,H2,MODEL)
      IF (.NOT.ISPOT) WRITE (IOOUT,4600)
      IF (.NOT.ISPOT) WRITE (IOOUT,4700)
C****  BEGINNING OF TRANSMITTANCE CALCULATIONS
 1300 CONTINUE
      IF (N16) KWAVE=KWAVE+1
      SUMV=0.
      TLOLD=1.
      TSOLD=1.
      TX7=1.
      TX10=1.
      IKLO=1
      TOLD=1
      IF (IEMISS.EQ.0) IKMAX=IKLO
C     ONLY ONE LOOP FOR TRANSMISSION: LOOP OVER LAYERS FOR EMISSION
      DO 2300 IK=IKLO,IKMAX
      IF (IEMISS.EQ.0) GO TO 1500
C     TRANSFER CUMULATIVE ABSORBER AMOUNTS FOR TH IK TH LEVEL AND
C     THE K TH ABSORBER - EMISSION ONLY.
      DO 1400 K=1,KMAX
      W(K)=WPATH(IK,K)
 1400 CONTINUE
 1500 IJ=IK
      IF (ICOUNT.EQ.0) GO TO 1600
      IF (ICOUNT.EQ.50) GO TO 1600
      GO TO 1700
 1600 ICOUNT=0
      IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,4800)
 1700 DO 1800 K=1,KMAX
      TX(K)=1.0
 1800 CONTINUE
      ICOUNT=ICOUNT+1
      V=FLOAT(IV)
      ALAM=1.E+04/V
      I=(IV-830)/20+1
      SUM4=0.
      SUM5=0.
      SUM6=0.
      SUM8=0.
      SUM11=0.
      CALL FREQSL(I,IV,W,TX)
      TX(9)=SUM4+SUM5+SUM8+SUM11+SUM6
      IF (TX(9).EQ.0.0) GO TO 2000
      IF (TX(9).LE.0.1) GO TO 1900
      IF (TX(9).GT.20.) GO TO 2100
      TX(9)=EXP(-TX(9))
      GO TO 2200
 1900 TX(9)=1.0-TX(9)+0.5*TX(9)*TX(9)
```

```
      GO TO 2200
 2000 TX(9)=1.0
      GO TO 2200
 2100 TX(9)=0.
 2200 TX(9)=TX(1)*TX(2)*TX(3)*TX(9)*TX(12)*TX(13)*TX(14)
C     AEROSOL COMPUTATIONS UNTIL LABEL 1
      IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 1
C
C    IF SPOT IS CALLING LT4M WITH ITYPE = 3, DO NOT INCLUDE AEROSOLS
C
      IF (ISPOT.AND.ITYPE.EQ.3) GO TO 1
      EXT55=3.914/VIS
C     UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE
C     PASS HORIZONTAL DIST IF ITYPE=1, SLANT DISTANCE IF ITYPE GT 1.
      IF (ITYPE.EQ.1) RNG=RANGE
      IF (ITYPE.EQ.2.AND.(H2.GT.H1).AND.(RANGE.GT..5/COS(ANGLE*CA)))
     1  RNG=.5/COS(ANGLE*CA)
      IF (ITYPE.EQ.2.AND.(H2.LT.H1).AND.
     1  (RANGE.GT..5/COS((180.-ANGLE)*CA)))
     2        RNG=.5/COS((180.-ANGLE)*CA)
      IF (ITYPE.EQ.3.AND.(RANGE.GT..5/COS(ANGLE*CA)))
     1        RNG=.5/COS(ANGLE*CA)
      IF (ITYPE.EQ.3.AND.RANGE.LT..0001)
     1        RNG=.5/COS(ANGLE*CA)
      ISLANT=ITYPE-1
C     CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL
      CALL XSCALE(ALAM,88.,EXT55,TX7,IERR,ISLANT,IHAZE,RNG,ANGLE)
      IF (IERR.EQ.1) RETURN
C     USER OPTIONS
      IF (IHAZE.EQ.7) TX7=EXP(-BETAEX*RANGE)
      IF (ISPOT.AND.IHAZE.EQ.8) TX7=EXP(-BE(KWAVE)*RANGE)
1     CONTINUE
      TX(9)=TX(9)*TX7
      IF (IV.GE.13000) TX(3)=TX(8)
      TNEW=TX(9)
      IF (IEMISS.EQ.0) GO TO 2500
C     COMPUTER LIMITS
      BBIK=0.0
      IF(ABS(1.43879*V/TBBY(IK)).LT.85.) BBIK=FF(TBBY(IK),V)
C     AEROSOL COMPUTATIONS UNTIL LABEL 2
      IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 2
C     FIND AEROSOL ABSORPTION IN DIFFERENT WAVELENGTH BANDS FROM EXTN
      IF (ALAM.LT.2.) TX10=1.
      IF (ALAM.GE.3..AND.ALAM.LE.5.) TX10=TX7**.2
      IF (ALAM.GE.8..AND.ALAM.LE.12.05) TX10=TX7**.45
2     CONTINUE
      TLNEW=TX(9)*TX10/(TX(6)*TX7)
      TSNEW=TX7*TX(6)/TX10
      DTAU=ABS(TLOLD-TLNEW)
      IF (DTAU.LT.1.0E-5.AND.TLNEW.LT.1.0E-5) GO TO 2400
      SUMV=SUMV+(TOLD-TNEW)*BBIK
      TLOLD=TLNEW
      TSOLD=TSNEW
      TOLD=TNEW
 2300 CONTINUE
 2400 CONTINUE
      TAUG=0.
      IF (HMIN.LE.0.0.AND.IL.EQ.1) TAUG=TX(9)
      T1=T(1)
      IF (TBOUND.GT.0.0) T1=TBOUND
C     COMPUTER LIMITS
      BBG=0.0
      IF(ABS(1.43879*V/T1).LT.85.) BBG=FF(T1,V)*TAUG
      IF (N16) RADG(KWAVE)=BBG*1.E+04
      IF (N16) RADA(KWAVE)=SUMV*1.E+04
      IF (HMIN.LE.0) SUMV=SUMV+BBG
      SUMVV=SUMV
      IF (IV.GT.IV1) FACTOR=1.0
      IF (IV.GE.IV2) FACTOR=0.5
      SUMV=(1.0E+04/V**2)*SUMV
```

```
      RADSUM=RADSUM+DV*FACTOR*SUMV
      IF (.NOT.ISPOT) WRITE (IOOUT,4900) V,
     1    ALAM,SUMV,SUMVV,RADSUM,TX(9),TX7,TX10
      IF (NPLT.EQ.1) WRITE (NPLOTU,98) ALAM,SUMVV
98    FORMAT (F7.4,1X,E13.5)
      IF (SUMV.GE.RADMAX) VRMAX=V
      IF (SUMV.GE.RADMAX) RADMAX=SUMV
      IF (SUMV.LE.RADMIN) VRMIN=V
      IF (SUMV.LE.RADMIN) RADMIN=SUMV
 2500 AB=1.-TX(9)
      IF (IV.EQ.IV1.OR.IV.GE.IV2) AB=0.5*AB
      SUMA=SUMA+AB*DV
      IF (IEMISS.EQ.1) GO TO 2600
      IF (.NOT.ISPOT) WRITE (IOOUT,5000) IV,
     1    ALAM,(TX(K),K=1,6),TX(11),TX(12),TX(13),TX(14),
     2    SUMA,TX(9),TX7
      IF (NPLT.EQ.1) WRITE(NPLOTU,99) ALAM,(TX(K),K=1,6),(TX(J),J=11,14)
99    FORMAT (F7.4,10(1X,F6.4))
 2600 CONTINUE
      RESPFN=RESFN(NR,ALAM)
      SUMRPF=SUMRPF+RESPFN
      IF (IV.GT.IV1) SUMINT=SUMINT+.5*(OLDTX9*OLDRFN+TX(9)*RESPFN)*
     +ABS(1./FLOAT(IV)-1./FLOAT(IV-IDV))*1.E+04
      IF (N16) TRAN(KWAVE)=TX(9)
      OLDTX9=TX(9)
      OLDRFN=RESPFN
      IV=IV+IDV
      IF (IV.GE.IV2) GO TO 2700
      GO TO 1300
 2700 CONTINUE
      IF (NR.NE.1) SUMRPF=1.
      SUMINT=SUMINT/(SUMRPF*1.E+04*ABS(1./FLOAT(IV1)-1./FLOAT(IV2)))
      IF (.NOT.ISPOT) TRAN(1)=SUMINT
      IF (.NOT.ISPOT.AND.NR.EQ.1) WRITE (IOOUT,3250) SUMINT
      IF (.NOT.ISPOT.AND.NR.NE.1) WRITE (IOOUT,3275) SUMINT
      RESPFN=0.
      SUMRPF=0.
      SUMINT=0.
      IF ((IEMISS.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,5100)
     1      VRMIN,RADMIN,VRMAX,RADMAX
      JSTOR=0
      AB=1.0-SUMA/FLOAT(IV-IV1)
      IF (ISPOT) RETURN
      WRITE (IOOUT,5200) IV1,IV,SUMA,AB
      IF ((IEMISS.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,5300) RADSUM
      IF (.NOT.ISPOT) READ (IOIN,3300) IXY
      IF (IXY.EQ.0) GO TO 3100
      GO TO (2800,700,2900,680,3100),IXY
 2800 CONTINUE
      READ (IOIN,5400) V1,V2,MULDV
      CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN(1))
      IF (IERR.EQ.1) RETURN
      AVW=10000./V1
      ALAM=10000./V2
      WRITE (IOOUT,5500) V1,V2,DV,ALAM,AVW
      SUMA=0.0
      GO TO 1100
 2900 IF (MODEL.EQ.0) GO TO 800
      GO TO 401
 3000 CONTINUE
      READ (IOIN,3300) MODEL,IHAZE,ITYPE,LEN,JP,NPLT,IM,
     1      ML,IEMISS,RO,TBOUND,BETAEX
      IF (IEMISS.EQ.1) WRITE (IOOUT,4100)
      IF (IEMISS.EQ.0) WRITE (IOOUT,4200)
      LENTOR=LEN
      GO TO 800
3100  RETURN
C
3250  FORMAT (/,1X,48HWAVELENGTH AND SENSOR INTEGRATED TRANSMISSION =
     +,E10.4)
```

```
3275  FORMAT (/,1X,37HWAVELENGTH INTEGRATED TRANSMISSION = ,E10.4)        L4M04220
 3300 FORMAT ( 9I3,3F10.3)                                                 L4M04230
 3500 FORMAT (F6.1,2(E9.3,F5.1,E9.3,2E7.1))                                L4M04240
 3510 FORMAT(3F8.3)                                                        L4M04250
 3550 FORMAT(F6.1,37X,E9.3,F5.1,E9.3,2E7.1)                                L4M04260
 3700 FORMAT (4(F6.3,2F7.4))                                               L4M04270
 3800 FORMAT (15F5.2)                                                      L4M04280
 3810 FORMAT(15F5.3)                                                       L4M04290
 3900 FORMAT (8E9.2)                                                       L4M04300
 4000 FORMAT (12F6.3)                                                      L4M04310
 4010 FORMAT(10F8.4)                                                       L4M04320
 4020 FORMAT(10F8.3)                                                       L4M04330
 4100 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,//,1X,       L4M04340
     1 45HPROGRAM WILL BE EXECUTED IN THE EMISSION MODE)                   L4M04350
 4200 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,//,1X,       L4M04360
     1 49HPROGRAM WILL BE EXECUTED IN THE TRANSMISSION MODE)               L4M04370
 4300 FORMAT (/10X,38H EQUIVALENT SEA LEVEL ABSORBER AMOUNTS               L4M04380
     1//21X,56HWATER VAPOUR      CO2 ETC.        OZONE     NITROGEN (CONT),L4M04390
     2        42H  H2O (CONT)        MOL  SCAT     OZONE(U-V)/24X,         L4M04400
     3        7HGM CM-2,10X,2HKM,10X,6HATM CM,10X,2HKM,9X,7HGM CM-2,       L4M04410
     4 10X,2HKM,10X,6HATM CM)                                              L4M04420
 4400 FORMAT(/10X,10H W(1-6,8)=,7(E14,3)/,1X,10X,7H W(10)=,58X,E14.3/)     L4M04430
 4401 FORMAT(//23X,11HNITRIC ACID,8X,3HSO2,11X,3HNH3,11X,3HNO2/)           L4M04440
 4500 FORMAT(/10X,10H W(11-16)=,6(E14,3)/)                                 L4M04450
 4600 FORMAT (1H1,30X,28HRADIANCE(WATTS/CM2-STER-XXX))                     L4M04460
 4700 FORMAT (1H ,10X,37HFR(CM-1)  WVL(MICRON)      PER CM-1    ,          L4M04470
     1        10HPER MICRON,26H      INTEGRAL          TRANS,1X,4(1H-),    L4M04480
     2      11H AERO TRAN ,4(1H-),/,1X,84X,17H EXTN            ABS)        L4M04490
 4800 FORMAT  (1H1,/,1X,2X,15HFREQ WAVELENGTH,2X,3HH2O,3X,4HCO2+,4X,       L4M04500
     1        30HOZONE   N2 C    H2O C   MOL S  ,1X,                       L4M04510
     2        22HNITRIC     SO2     HNO3,4X,16HNO2    INTEGRATED,          L4M04520
     3        2X,13HTOTAL AEROSOL/1X,1X,13H CM-1 MICRONS,10(3X,5HTRANS),L4M04530
     4        2X,24HABSORPTION  TRANS  TRANS)                              L4M04540
 4900 FORMAT (1H ,10X,F8.1,F13.6,3E13.5,F13.6,1X,F7.5,3X,F7.5)             L4M04550
 5000 FORMAT (1H ,I6,11F8.4,F11.4,F8.4,1X,F7.5)                            L4M04560
 5100 FORMAT (1H0,8H RADMIN ,F12.3,E12.5,/,8H RADMAX ,F12.3,               L4M04570
     1        E12.5)                                                       L4M04580
 5200 FORMAT (1H0,26H INTEGRATED ASORPTION FROM,I5,4H TO ,I5,              L4M04590
     1        7H CM-1 =,F10.2,25H, AVERAGE TRANSMITTANCE =,F6.4)           L4M04600
 5300 FORMAT (1H ,22H INTEGRATED RADIANCE =,E12.5,13H WATT CM -2  ,2HSR)L4M04610
 5400 FORMAT (2F10.3,I2)                                                   L4M04620
 5500 FORMAT (/10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO ,              L4M04630
     1        4HV2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1   (               L4M04640
     2        ,F6.2,3H - ,F5.2,10H MICRONS ))                              L4M04650
      END                                                                  L4M04660
```

```
      SUBROUTINE ABSORB(IXY,IERR,W,V1,V2,DV,SUMA,MULDV,ANGLE,LEN,ITYPE,
     1          H1,H2,MODEL,ISPOT1,RANGE,BETA,VIS,ICLMAT,IV1,IV2,IDV)
      COMMON /GEOMET/PTS(15),IGEOSW
      COMMON /CLYMAT/TEMP,PRESS,RH1,AH1,DP1,VIS1,CLDAMT,
     1       CLDHYT,FOGPRB,WNDVEL,WNDDIR,IPASCT
      COMMON /CONST/PI,PI2,CA,TWOPI,TORRMB,CDEGK
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NFLOTU
      COMMON /M01/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL
      COMMON /M02/ WO(34),RO,TBOUND,JP,IM,ML,IP,JSTOR
      COMMON /M09/RADMAX,RADMIN,VRMAX,VRMIN
      COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1
      COMMON /LOWEX/WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX,
     1 CLDHGT,NCLD
      COMMON /SPOTLO/ISPOT,LOREAD,N16
      COMMON /M03/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),
     1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2
      LOGICAL ISPOT,N16,LOREAD
      DIMENSION VH(16),W(16),E(16)
C     EH(7,I) REPLACES HSTOR
C     EH(9,I) REPLACES HMIX
      DATA (EH(9,I),I=1,34) /9*0.,0.1,0.33,0.8,1.2,1.4,1.6,1.8,1.9,
     1 2.0,2.1,2.3,3.0,3.7,4.2,5.2,6.0,3.8,2.6,0.22,6*0.0/
      F(A)=EXP(18.9766-14.9595*A-2.43882*A*A)*A
      TMPVIS=VIS
      IF (ISPOT1.EQ.1) GO TO 200
      IF (MODEL.EQ.0) GO TO 400
      IF (IXY.EQ.3) GO TO 100
      IF (M.EQ.7.AND.IM.NE.0) GO TO 400
      IF (IXY.GT.3) GO TO 1500
C     WHEN IXY=0 VIS IS READ IN MAIN
  100 IF (.NOT.ISPOT) READ (IOIN,6200) H1,H2,ANGLE,RANGE,
     1     BETA,VIS,CLDHGT
      IF(IGEOSW.NE.1) GO TO 111
      H1=PTS(3)
      H2=PTS(6)
      RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+
     +(PTS(3)-PTS(6))**2)
  111 CONTINUE
      IF (IXY.EQ.0) VIS=TMPVIS
  200 X1=RE+H1
      X2=RE+H2
      IF (ITYPE.EQ.3) GO TO 1000
      IF (ITYPE.EQ.1) GO TO 1500
      IF (RANGE.EQ.0.) GO TO 1200
      IF (.NOT.ISPOT) WRITE (IOOUT,6300) H1,H2,ANGLE,RANGE,
     1     BETA,VIS
      IF (H2.EQ.0.AND.ANGLE.NE.0) GO TO 300
      ANGLE= ACOS(0.5*((H2-H1)*(1.+X2/X1)/RANGE-RANGE/X1))/CA
      GO TO 1400
  300 X2=SQRT((X1/RANGE+RANGE/X1+2.0*COS(ANGLE*CA))*X1*
     1          RANGE)
      H2=X2-RE
      GO TO 1400
  400 CONTINUE
      IF (ML.LE.0) ML=1
      DO 900 K=1,ML
C     CLIMATE OPTION - SEE COMMON /CLYMAT/
      IF (M.EQ.0.AND..NOT.ISPOT) READ (IOIN,6400) H1,P(1),
     1     TMP,DP,RH,WH(K),WO(K),VIS,RANGE
      IF(IGEOSW.NE.1) GO TO 444
      H1=PTS(3)
      H2=PTS(6)
      RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+
     +(PTS(3)-PTS(6))**2)
  444 CONTINUE
      IF (ICLMAT.NE.1) GO TO 500
      TMP=TEMP
      P(1)=PRESS
      DP=DP1
      RH=RH1
```

```
      WH(K)=AH1
      VIS=VIS1
  500 IF (IXY.EQ.0.AND.ICLMAT.NE.1) VIS=TMPVIS
      IF ((M.GT.0).AND.(.NOT.ISPOT)) READ (IOIN,6400) Z(K),
     1    P(K),TMP,DP,RH,WH(K),WO(K)
      IF ((M.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,6500) H1,
     1    P(1),TMP,DP,RH,WH(K),WO(K),VIS,RANGE
      IF (M.EQ.0) Z(K)=H1
      J=IFIX(Z(K)+1.0E-6)+1
      IF (Z(K).GE.25.) J=(Z(K)-25.)/5.0+26.
      IF (Z(K).GE.50.0) J=(Z(K)-50.)/20.+31.
      IF (Z(K).GE.70.) J=(Z(K)-70.)/30.+32.
      IF (J.GT.33) J=33
      FAC=Z(K)-FLOAT(J-1)
      IF (J.LT.26) GO TO 600
      FAC=(Z(K)-5.0*FLOAT(J-26)-25.)/5.
      IF (J.GE.31) FAC=(Z(K)-50.)/20.
      IF (J.GE.32) FAC=(Z(K)-70.)/30.
      IF (FAC.GT.1.0) FAC=1.0
  600 L=J+1
      T(K)=TMP+CDEGK
      TT=CDEGK/T(K)
      IF (RH.LE.0.0) TT=CDEGK/(CDEGK+DP)
      IF (WH(K).LE.0.0) WH(K)=F(TT)
      IF (RH.GT.0.0) WH(K)=0.01*RH*WH(K)
      EH(7,K)=0.0
      IF (EH(9,J).LE.0.) GO TO 700
      EH(7,K)=EH(9,J)*(EH(9,L)/EH(9,J))**FAC
  700 CONTINUE
      IF (MODEL.EQ.0) GO TO 1500
      IF ((K.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,6600)
      IF (.NOT.ISPOT) WRITE (IOOUT,6400) Z(K),P(K),TMP,DP,
     1    RH,WH(K),WO(K)
  900 CONTINUE
      IM=0
      NL=ML
C     NOTE THAT Z(I) MAY NOT CORRESPOND TO THE VALUES GIVEN FOR STANDARD
C     MODEL ATMOSPHERES
      IF (IXY.GE.3) GO TO 1500
      GO TO 100
 1000 IF (RANGE.GT.0.0) GO TO 1100
      GO TO 1500
 1100 ITYPE=2
      BETA= ACOS(0.5*(RANGE*RANGE/(X1*X2)-X2/X1-X1/X2))/CA
 1200 IF (BETA.EQ.0.) GO TO 1300
      BET=CA*BETA
      X2=RE+H2
      ANGLE=ATAN(X2*SIN(BET)/(X2*COS(BET)-X1))/CA
      IF (ANGLE.LT.0.) ANGLE=ANGLE+PI
      RANGE=X2*SIN(BET)/SIN(ANGLE*CA)
      BET=BETA
      GO TO 1500
 1300 RANGE=(X2/X1)**2-(SIN(ANGLE*CA))**2
      IF (RANGE.GE.0.0) RANGE=X1*(SQRT(RANGE)-ABS(COS(ANGLE*CA)))
 1400 IF (ANGLE.NE.0..OR.ANGLE.NE.180.) BET=ASIN(RANGE*SIN(ANGLE*CA)/X2)
      IF (ANGLE.LT.0.) ANGLE=ANGLE+PI
      IF (RANGE.LT.0.0) RANGE=-RANGE
      BET=BET/CA
      IF (.NOT.ISPOT) WRITE (IOOUT,6300) H1,H2,ANGLE,RANGE,
     1    BET,VIS
 1500 CONTINUE
      DO 1600 I=1,NL
      DO 1600 J=1,KMAX
 1600 WLAY(I,J)=0.
      SUMA=0.
C     WHEN IXY=0 V1,V2,MULDV ARE READ IN EOSAEL.MAIN
      IF ((IXY.EQ.1.OR.IXY.EQ.2).AND.(.NOT.ISPOT))
     1     READ (IOIN,6250) V1,V2,MULDV
      IF ((IXY.EQ.1.OR.IXY.EQ.2).AND.(.NOT.ISPOT))
     + CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TMPVIS)
```

```
      IF (IERR.EQ.1) RETURN
      IF ((ITYPE.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,6700)
     1     H1,RANGE
      IF ((ITYPE.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,6800)
     1     H1,H2,ANGLE
      IF ((ITYPE.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,6900)
     1     H1,ANGLE
      IF (MODEL.EQ.0) M=7
      IF ((M.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,7200) M
      IF ((M.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,7300) M
      IF ((M.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,7400) M
      IF ((M.EQ.4).AND.(.NOT.ISPOT)) WRITE (IOOUT,7500) M
      IF ((M.EQ.5).AND.(.NOT.ISPOT)) WRITE (IOOUT,7600) M
      IF ((M.EQ.6).AND.(.NOT.ISPOT)) WRITE (IOOUT,7700) M
      IF ((M.EQ.8).AND.(.NOT.ISPOT)) WRITE (IOOUT,7800) M
      IF ((M.EQ.9).AND.(.NOT.ISPOT)) WRITE (IOOUT,7900) M
      AVW=10000./V1
      ALAM=10000./V2
      RADMIN=1.0E+38
      RADMAX=0.
      VRMIN=0.
      VRMAX=0.
      IF (.NOT.ISPOT) WRITE (IOOUT,8000) V1,V2,DV,ALAM,AVW
      AVW=0.5E-4*(V1+V2)
      AVW=AVW*AVW
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8100)
      IF (ITYPE.EQ.1) GO TO 2100
      DO 1800 K=1,KMAX
      VH(K)=0.0
 1800 CONTINUE
      BETA=0.0
      SR=0.0
      IP=0
C****  NOW DEFINE CONSTANT PRESSURE PATH QUANTITES  EH(1-8)
      Y=CA*ANGLE
      SPHI=SIN(Y)
      R1=(RE+H1)*SPHI
      IF (H1.GT.Z(NL)) GO TO 1900
      GO TO 2100
 1900 X=(RE+Z(NL))/(RE+H1)
      IF (SPHI.GT.X) GO TO 2000
      H1=Z(NL)
      J1=NL
      SPHI=SPHI/X
      ANGLE=180.0- ASIN(SPHI)/CA
      R1=(RE+H1)*SPHI
      GO TO 2100
 2000 HMIN=R1-RE
      IF (.NOT.ISPOT) WRITE (IOOUT,8200) HMIN
      GO TO 6000
 2100 DO 2400 I=1,NL
      PS=P(I)/1013.0
      TS=CDEGK/T(I)
      X=PS*TS
C--- COMPUTE MASS DENSITY (G M-3) FROM IDEAL GAS LAW ---
C--- 1292.02 = DENSITY OF STANDARD COMPOSITION AIR AT STP ---
      WA(I) = 1292.02*X
      PT=PS*SQRT(TS)
      D=0.1*WH(I)
      EH(1,I)=D*PT**0.9
      EH(2,I)=X*PT**0.75
      EH(4,I)=0.8*PT*X
      PPW=4.56E-5*D*CDEGK/TS
      TS1=(296.0/CDEGK)*TS
      EH(5,I)=D*PPW*EXP(6.08*(TS1-1.0))+0.002*D*(PS-PPW)
      EH(10,I)=D*(PPW+0.12*(PS-PPW))*EXP(4.56*(TS1-1.0))
      EH(6,I)=X
      EH(8,I)=46.6667*WO(I)
      EH(3,I)=EH(8,I)*PT**0.4
C EH(11,I)=HNO3 ABSORBER AMOUNT (ATM-CM)/KM
```

```
      EH(11,I)=PS*TS*EH(9,I)*1.0E-04
      IF (MODEL.EQ.0.OR.MODEL.EQ.7) EH(11,I)=PS*TS*EH(7,I)*1.0E-04
C*****EH(12,I) = SO2   ABSORBER AMOUNT (ATM-CM)/KM
      EH(12,I)=0.772E-04*PPMSO2*WA(I)*PS**0.07122*TS**0.06159
      EH(13,I)=0.772E-04*PPMNH3*WA(I)*PS**0.52125*TS**(-0.60438)
      EH(14,I)=0.772E-04*PPMNO2*WA(I)*PS**0.18066*TS**0.20911
C***K=15 FOR ASL 3.3 - 4.3_ MICRON H2O CONTINUUM
      EH(15,I)=PPW*D
C***K=16 FOR 4.6 - 4.8 MICRON H2O CONTINUUM
      EH(16,I)=PPW*(PPW+3.0E-03*(PS-PPW))
      IF (I.EQ.NL) GO TO 2300
      IF (MODEL.EQ.0.AND.I.GE.1) GO TO 3600
      T2=T(I+1)
      W2=WH(I+1)
      PPW=4.56E-6*W2*T2
 2300 IF (H1.GE.Z(I)) J1=I
      IF ((JP.EQ.0).AND.(.NOT.ISPOT))
     1WRITE (IOOUT,8300)I,Z(I),(EH(K,I),K=1,6),EH(8,I),(EH(K,I),K=10,14)
 2400 CONTINUE
      X1=H1
      CALL POINT (H1,N,NP1,TX)
      J1=N
      DO 2500 K=1,KMAX
 2500 E(K)=TX(K)
      JEXTRA=0
      JMIN=0
C**ITYPE=1 MEANS HORIZONTAL PATH ***************************************
      IF (ITYPE.EQ.1) GO TO 3600
      IF (ITYPE.EQ.3) H2=Z(NL)
C** ANGLE GREATER THAN 90 DEGREES MEANS DOWNWARD TRAJECTORY ************
      IF (ANGLE.GT.90.0) GO TO 3800
C** IF THE PATH IS NOT HORIZONTAL OR DOWNWARD THEN IT IS UPWARD TRAJECTO
 2600 IF (ANGLE.GT.90.0.AND.NP1.GT.0) J1=J1+1
      J2=NL
      IF (ITYPE.EQ.3) GO TO 2700
      CALL POINT (H2,N,NP,TX)
      J2=N
      IF (NP.GT.0) J2=J2-1
 2700 DO 2800 K=1,KMAX
      IF (K.EQ.9.OR.K.EQ.7) GO TO 2800
      EH(K,J1)=E(K)
      IF (ITYPE.EQ.3) GO TO 2800
      EH(K,J2+1)=TX(K)
2800  CONTINUE
C****  NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8400)
      DO 2900 K=1,KMAX
 2900 W(K)=0.
      DO 3500 I=J1,J2
      X1=Z(I)
      IF (I.LT.NL) X2=Z(I+1)
      IF (I.EQ.NL) X2=Z(I)
      IF (I.EQ.J1) X1=H1
      IF (I.EQ.J2) X2=H2
      DZ=X2-X1
      IF (I.EQ.NL) DZ=Z(I)-Z(I-1)
      DS=DZ
C*****  UPWARD TRAJECTORY
      RX=(RE+X1)/(RE+X2)
      THETA= ASIN(SPHI)/CA
      PHI= ASIN(SPHI*RX)/CA
      BET=THETA-PHI
      SALP=RX*SPHI
      IF (SPHI.GT.1.E-10) DS=(RE+X2)*SIN(BET*CA)/SPHI
      BETA=BETA+BET
      PSI=BETA+PHI-ANGLE
      PHI=180.-PHI
      SR=SR+DS
      JEXTRA=0
      DO 3400 K=1,KMAX
```

```
      IF (K.EQ.7.OR.K.EQ.9) GO TO 3100
      EV=DS*EH(K,I)
      IF (I.EQ.NL) GO TO 3000
      IF (EH(K,I).EQ.0.0.OR.EH(K,I+1).EQ.0.0) GO TO 3100
      IF (EH(K,I).EQ.EH(K,I+1)) GO TO 3200
      EV=DS*(EH(K,I)-EH(K,I+1))/ALOG(EH(K,I)/EH(K,I+1))
      GO TO 3200
 3000 IF (EH(K,I).EQ.0.0) GO TO 3100
      IF (EH(K,I-1).EQ.0.0) GO TO 3100
      IF (EH(K,I).EQ.EH(K,I-1)) GO TO 3200
      EV=EV/ALOG(EH(K,I-1)/EH(K,I))
      GO TO 3200
 3100 EV=0.
 3200 VH(K)=VH(K)+EV
      IF (I.EQ.JSTOR) GO TO 3300
      WLAY(I,K)=EV+W(K)
      W(K)=0.
      GO TO 3400
 3300 W(K)=EV
      IF (J1.NE.J2) GO TO 3400
      WLAY(J2+1,K)=W(K)
      W(K)=0.
      JEXTRA=1
 3400 CONTINUE
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8500) I,
     1   X1,(VH(L),L=1,6),VH(8),PSI,PHI,BETA,THETA,SR
      IF (I.GE.NL) GO TO 3500
      SPHI=SPHI*RX
      IF (SALP.GE.1.) SPHI=SALP
3500  CONTINUE
      GO TO 5800
C**** HORIZONTAL PATH
 3600 DO 3700 K=1,KMAX
      IF (K.EQ.7.OR.K.EQ.9) GO TO 3700
      W(K)=RANGE*EH(K,1)
      IF (MODEL.GT.0) W(K)=RANGE*TX(K)
      VH(K)=W(K)
 3700 CONTINUE
      GO TO 6100
 3800 CONTINUE
C****   DOWNWARD TRAJECTORY
      K2=0
      IF (NP1.EQ.1) J1=J1-1
      IF (J1.LE.0) J1=1
      J2=J1+1
      J=J1+1
      IF (H2.GT.Z(J1+1).OR.H1.EQ.H2) GO TO 4000
      IF (NP1.EQ.1.AND.H2.GE.Z(J1+1)) GO TO 4000
      CALL POINT (H2,N,NP2,TX)
      DO 3900 K=1,KMAX
 3900 W(K)=TX(K)
      IF (H2.LT.H1) H=H2
      J2=N
 4000 A0=(RE+H1)*SPHI
      DO 4100 I=1,J1
      HMIN=A0-RE
      JMIN=I
      IF (HMIN.LE.Z(I+1)) GO TO 4200
 4100 CONTINUE
 4200 X=HMIN
      IF (HMIN.LE.0) GO TO 4400
      CALL POINT (X,N,NP,TX)
      JMIN=N
      HMIN=A0-RE
      IF (ABS(X-HMIN).GT.0.0001) GO TO 4200
      IF (H2.GE.H1) J2=N
      IF (H2.GE.H1.OR.H2.LT.HMIN) H=HMIN
      IF (.NOT.ISPOT) WRITE (IOOUT,8600) HMIN
      IF (H2.LT.HMIN) J2=N
      IF ((H2.LT.HMIN).AND.(.NOT.ISPOT)) WRITE (IOOUT,8700) HMIN
```

```
      GO TO 4500
 4400 IF (.NOT.ISPOT) WRITE (IOOUT,8600) HMIN
      IF (H2.LT.H1) GO TO 4500
      IF ((ITYPE.EQ.3.OR.H2.GE.H1).AND.(.NOT.ISPOT))
     1     WRITE (IOOUT,8800)
      ITYPE=2
      JMIN=0
      J2=1
      H2=0.0
      H=0.0
C****  NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)
 4500 IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8400)
      JSTOR=J-1
      DO 5100 I=1,NL
      J=J-1
      IF (I.NE.1) X1=Z(J+1)
      X2=Z(J)
      IF (J.EQ.J2.AND.K2.EQ.0) X2=H
      IF (J.EQ.JMIN.AND.K2.EQ.1) X2=HMIN
      HM=(RE+X1)*SPHI-RE
      IF (HM.GT.Z(J).AND.HM.GT.X2) X2=HM
      RX=(RE+X1)/(RE+X2)
      DS=X1-X2
      ALP=90.0
      THET= ASIN(SPHI)/CA
      SALP=RX*SPHI
      IF (ABS(2-HM).GT.1.0E-5) ALP= ASIN(SALP)/CA
      BET=ALP-THET
      IF (SPHI.GT.1.0E-10) DS=(RE+X2)*SIN(BET*CA)/SPHI
      THETA=180.0-THET
      BETA=BETA+BET
      PSI=BETA-ALP-ANGLE+180.0
      SR=SR+DS
      DO 5000 K=1,KMAX
      IF (K.EQ.7.OR.K.EQ.9) GO TO 5000
      AJ=EH(K,J)
      BJ=EH(K,J+1)
      IF (J.EQ.J1) BJ=E(K)
      IF (J.EQ.J2.AND.H2.LT.H1.AND.H2.GT.0.0) AJ=W(K)
      IF (J.EQ.JMIN.AND.H2.GE.H1) AJ=TX(K)
      IF (J.EQ.JMIN.AND.ABS(H2-HM).LT.1.0E-5) AJ=TX(K)
      IF (K2.EQ.0) GO TO 4600
      IF (J.EQ.J2) BJ=W(K)
      IF (J.EQ.JMIN) AJ=TX(K)
 4600 IF (AJ.EQ.0.0.OR.BJ.EQ.0.0) GO TO 4800
      IF (AJ.EQ.BJ) GO TO 4700
      EV=DS*(AJ-BJ)/ALOG(AJ/BJ)
      GO TO 4900
 4700 EV=DS*AJ
      GO TO 4900
 4800 EV=0.0
 4900 VH(K)=VH(K)+EV
 5000 WLAY(J,K)=EV
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8500) J,
     1     X1,(VH(L),L=1,6),VH(8),PSI,ALP,BETA,THETA,SR
      IF (J.EQ.J2.AND.H2.GE.H1) GO TO 5600
      IF (J.EQ.JMIN.AND.K2.EQ.1) GO TO 5400
      SPHI=SALP
      IF (J.EQ.J2.AND.K2.EQ.0) GO TO 5200
 5100 CONTINUE
 5200 IF (HMIN.LE.0) GO TO 5800
      IF ((LEN.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8900)
      IF ((LEN.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,9000)
      IF (LEN.EQ.0) GO TO 5800
      K2=1
      X1=X2
      IF (ABS(X1-HMIN).LE.0.001) GO TO 5800
      H=HMIN
      J=J2+1
      IF (NP2.EQ.1) J=J-1
```

```
      B=BETA
      PH=180.0- ASIN(SPHI)/CA
      TS=SR
      PS=PSI
      DO 5300 K=1,KMAX
 5300 E(K)=VH(K)
      GO TO 4500
 5400 BETA=2.*BETA-B
      PSI=2.*PSI-PS
      SR=2.*SR-TS
C       LONG PATH  TAKEN
      PHI=PH
      DO 5500 K=1,KMAX
 5500 VH(K)=2.*VH(K)-E(K)
      GO TO 5800
 5600 DO 5700 K=1,KMAX
 5700 VH(K)=2.0*VH(K)
      BETA=2.0*BETA
      SR=2.0*SR
      IF (H2.EQ.H1) GO TO 5800
      SPHI=SIN(ANGLE*CA)
      GO TO 2600
 5800 CONTINUE
      DO 5900 K=1,KMAX
      W(K)=VH(K)
 5900 CONTINUE
 6000 CONTINUE
 6100 RETURN
C
 6200 FORMAT (7F10.3)
 6250 FORMAT (2F10.3,I2)
 6300 FORMAT (1H0,9X,4H H1=,F7.3,6HKM,H2=,F7.3,9HKM,ANGLE=,
     1          F8.4,13HGEOM. RANGE =,F7.2,8HKM,BETA=,F8.5,
     2          5H,VIS=,F6.1)
 6400 FORMAT (3F10.3,2F5.1,2E10.3,2F10.3)
 6500 FORMAT (10X,26HINPUT METEOROLOGICAL DATA:/10X,2HZ=,
     1          F7.2,7H KM, P=,F7.2,6H MB,T=,F5.1,8H C, DEW ,
     2          7HPT.TEMP,F5.1,17H C, REL HUMIDITY=,F5.1,
     3          16H %, H2O DENSITY=,1PE9.2,7H GM M-3/10X,
     4          15H OZONE DENSITY=,E9.2,20H GM-3, VISUAL RANGE=
     5          ,0PF6.1,10H KM,RANGE=,F10.3,4H KM  )
 6600 FORMAT (24H  MODEL ATMOSPHERE NO. 7,/4X,6HZ (KM),3X,
     1          6HP (MB),4X,30HT (C) DEW PT %RH  H2O(GM.M-3) ,
     2          19HO3(GM.M-3) NO. DEN.)
 6700 FORMAT (//10X,28H HORIZONTAL PATH, ALTITUDE =,F7.3,
     1          11H KM,RANGE =,F7.3,3H KM)
 6800 FORMAT (//10X,37H SLANT PATH BETWEEN ALTITUDES H1 AND ,
     1          13HH2 WHERE H1 =,F7.3,8H KM H2 =,F7.3,
     2          18H KM,ZENITH ANGLE =,F7.3,8H DEGREES)
 6900 FORMAT (//10X,39H SLANT PATH TO SPACE FROM ALTITUDE H1 =
     1          ,F7.3,19H KM, ZENITH ANGLE =,F7.3,
     2          8H DEGREES)
 7200 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,11H = TROPICAL)
 7300 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,
     1          21H = MIDLATITUDE SUMMER)
 7400 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,
     1          21H = MIDLATITUDE WINTER)
 7500 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,14H = SUB-ARCTIC ,
     1          7HSUMMER )
 7600 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,14H = SUB-ARCTIC ,
     1          7HWINTER )
 7700 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,11H = 1962 US ,
     1          10HSTANDARD   )
 7800 FORMAT(/20X,18H MODEL ATMOSPHERE ,I1,20H = ISRAELI STANDARD ,
     1          16H(YEAR, DAYTIME) )
 7900 FORMAT(/20X,18H MODEL ATMOSPHERE ,I1,20H = ISRAELI STANDARD ,
     1          18H(YEAR, NIGHTTIME) )
 8000 FORMAT (/10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO ,
     1          4HV2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1    (
     2          ,F6.2,3H - ,F5.2,10H MICRONS ))
```

```
8100 FORMAT (1H1,///10X,20H HORIZONTAL PROFILES/)                       ABS04910
8200 FORMAT (38H TRAJECTORY MISSES EARTHS ATMOSPHERE. ,                 ABS04920
    1        31HCLOSEST DISTANCE OF APPROACH IS,F10.2,1X,/              ABS04930
    2        ,1X,18HEND OF CALCULATION)                                 ABS04940
8300 FORMAT (1X,I4,F6.1,12(E9.3))                                       ABS04950
8400 FORMAT (1H1,///10X,21H  VERTICAL PROFILES   ,53X,3HPSI,            ABS04960
    1        6X,3HPHI,6X,4HBETA,3X,13HTHETA    RANGE)                   ABS04970
8500 FORMAT (I3,F6.1,7E10.3,4F9.4,F6.1)                                 ABS04980
8600 FORMAT (8H HMIN = ,F10.3)                                          ABS04990
8700 FORMAT (40H H2 WAS SET LESS THAN HMIN AND HAS BEEN ,               ABS05000
    1        34HRESET EQUAL TO      HMIN I.E. H2 = ,F10.3)              ABS05010
8800 FORMAT (41H PATH INTERSECTS EARTH - PATH CHANGED TO ,              ABS05020
    1        23HTYPE 2 WITH H2 = 0.0 KM)                                ABS05030
8900 FORMAT (36H CHOICE OF TWO PATHS FOR THIS CASE -,                   ABS05040
    1        42HSHORTEST PATH TAKEN.  FOR LONGER PATH SET ,             ABS05050
    2        6HLEN=1.)                                                  ABS05060
9000 FORMAT (44H CHOICE OF TWO PATHS FOR THIS CASE -LONGEST ,           ABS05070
    1        41HPATH TAKEN.   FOR SHORT PATH SET LEN = 0 )              ABS05080
     END                                                                ABS05090
```

```
      SUBROUTINE CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN)
      LOGICAL ISPOT
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      IERR=0
      IF (MULDV.LE.0) MULDV=1
C     CHECK FOR PROPER WINDOW REGION
      IF(V2.LE.830.0) GO TO 1090
      IF (V1.GE.1250.0.AND.V2.LE.2010.0) GO TO 1090
      IF (V1.GE.3330.0.AND.V2.LT.5010.0) GO TO 1090
      IF (V1.GE.39990.0) GO TO 1090
      GO TO 1100
 1090 WRITE(IOOUT,3190)
      TRAN=1.
      IERR=1
      RETURN
C     CHECK FOR PROPER INTEGER VALUES
1100  IV1=20*IFIX((V1-10.0)/20.0)+10
      IV2=20*IFIX((V2-10.0)/20.0+0.99)+10
      IF (IV1.LT.830) IV1=830
      IF (IV1.LT.1250.AND.IV2.GT.1250) IV2=1250
      IF (IV1.LT.2010.AND.IV2.GT.2010) IV1=2010
      IF (IV1.LT.3330.AND.IV2.GT.3330) IV2=3330
      IF (IV1.LT.5010.AND.IV2.GT.5010) IV1=5010
      IF (IV1.LT.39990.AND.IV2.GT.39990) IV2=39990
      V1=FLOAT(IV1)
      V2=FLOAT(IV2)
      DV=20.*FLOAT(MULDV)
      IWVCK=0
      IDV=IFIX(DV)
C     WHEN CALLED FROM SPOT CHECK FOR MORE THAN 15 DIVISIONS
C     (16 WAVENUMBER VALUES) WHICH IS ARRAY SIZE.
C     CHECK HERE FOR ROUNDING PROBLEMS ON V1, V2 CAUSING
C     TOO SMALL AN INCREMENT
      IF (ISPOT.AND.(FLOAT(IV2-IV1)/15..GT.(FLOAT(IDV)+.001))) IWVCK=1
      IF (IWVCK.NE.1) GO TO 91
      CKDV=FLOAT(IV2-IV1)/(15.*20.)
      MULDV=IFIX(CKDV)
      IF (FLOAT(MULDV)/CKDV.LT..99) MULDV=MULDV+1
      DVHOLD=20.*FLOAT(MULDV)
      WRITE (IOOUT,93) DV,DVHOLD
      DV=DVHOLD
      IF (DV.LT.20.) DV=20.
      IDV=IFIX(DV)
93    FORMAT (1X,'DIVISION LIMITS CHANGED FROM ',F10.3,
     +' TO ',F10.3)
91    CONTINUE
      RETURN
 3190 FORMAT (6X,'******FREQUENCY IS OUTSIDE OF THE WINDOW******'/
     1         6X,'******TOTAL TRANSMITTANCE IS 1.0000******')
      END
```

```
      SUBROUTINE FREQSL(I,IV,W,TX)
      COMMON /M05/ C1(501),C2(258),C3(86),C4(33),C5(6),C5DUM(9),C8(82),
     1        C11(4),C12(15),C14(21),C15(6)
      COMMON /M07/ TR(67),FW(67),FO(67)
      COMMON /M08/SUM4,SUM5,SUM8,SUM11,SUM6
      DIMENSION W(16),TX(16)
      IF (I.EQ.1) GO TO 10
      IF (I.GE.2.AND.I.LE.3) GO TO 11
      IF (I.GE.4.AND.I.LE.5) GO TO 12
      IF (I.GE.6.AND.I.LE.12) GO TO 13
      IF (I.GE.13.AND.I.LE.21) GO TO 15
      IF (I.EQ.22) GO TO 16
      IF ((I.GE.23.AND.I.LE.59).OR.(I.GE.127.AND.I.LE.209)) RETURN
      IF ((I.GE.60.AND.I.LE.63)) GO TO 14
      IF (I.GE.64.AND.I.LE.76) GO TO 18
      IF ((I.GE.77.AND.I.LE.81).OR.(I.GE.87.AND.I.LE.96)) GO TO 18
      IF (I.GE.82.AND.I.LE.86) GO TO 30
      IF ((I.GE.97.AND.I.LE.101).OR.(I.GE.105.AND.I.LE.109)) GO TO 14
      IF (I.GE.102.AND.I.LE.104) GO TO 9
      IF (I.GE.110.AND.I.LE.112) GO TO 21
      IF (I.GE.113.AND.I.LE.123) GO TO 22
      IF ((I.GE.124.AND.I.LE.126).OR.(I.GE.210.AND.I.LE.363)) GO TO 23
      IF ((I.GE.364.AND.I.LE.419).OR.(I.GE.454.AND.I.LE.599)) GO TO 24
      IF ((I.GE.420.AND.I.LE.453).OR.(I.GE.600.AND.I.LE.606).OR.(I.GE
     1.1160.AND.I.LE.1334)) GO TO 35
      IF (I.GE.607.AND.I.LE.609) GO TO 25
      IF (I.GE.610.AND.I.LE.621) GO TO 26
      IF ((I.GE.622.AND.I.LE.629).OR.(I.GE.686.AND.I.LE.1159).OR.(I.GE
     1.1335)) GO TO 27
      IF (I.GE.630.AND.I.LE.685) GO TO 28
 4    CALL H2OVAP(I,W,C1,TX)
      GO TO 40
 6    CALL OZONE(I,W,C3,TX)
 7    CALL UNIMIX(I,W,C2,TX)
      GO TO 4
 8    CALL NO2(I,W,C15,TX)
      GO TO 6
 9    CALL H20410(I,IV,W,C5,TX,SUM5)
      GO TO 8
 10   CALL NH3(I,W,C14,TX)
      GO TO 9
 11   CALL NITRIC(I,W,C11,SUM11,TX)
      GO TO 10
 12   CALL NITRIC(I,W,C11,SUM11,TX)
 13   CALL NH3(I,W,C14,TX)
 14   CALL H20410(I,IV,W,C5,TX,SUM5)
      GO TO 6
 15   CALL SO2(I,W,C12,TX)
      GO TO 13
 16   CALL SO2(I,W,C12,TX)
      GO TO 14
 18   CALL H20410(I,IV,W,C5,TX,SUM5)
 19   CALL NITRO(I,W,C4,TX,SUM4)
      GO TO 6
 20   CALL SO2(I,W,C12,TX)
      GO TO 19
 21   CALL MOLSCT(IV,W,TX,SUM6)
      GO TO 14
 22   CALL MOLSCT(IV,W,TX,SUM6)
      GO TO 6
 23   CALL MOLSCT(IV,W,TX,SUM6)
      GO TO 7
 24   CALL MOLSCT(IV,W,TX,SUM6)
      GO TO 4
 25   CALL UNIMIX(I,W,C2,TX)
 35   CALL MOLSCT(IV,W,TX,SUM6)
      GO TO 40
 26   CALL UVOZNE(I,W,C8,TX,SUM8)
      GO TO 25
 27   CALL UVOZNE(I,W,C8,TX,SUM8)
```

```
      CALL MOLSCT(IV,W,TX,SUM6)                                         FRE00710
      GO TO 40                                                          FRE00720
28    CALL H2OVAP(I,W,C1,TX)                                            FRE00730
      GO TO 27                                                          FRE00740
30    CALL H20410(I,IV,W,C5,TX,SUM5)                                    FRE00750
      GO TO 20                                                          FRE00760
40    RETURN                                                            FRE00770
      END                                                               FRE00780
```

```
      SUBROUTINE H2OVAP(I,W,C1,TX)
      DIMENSION C1(501),TX(1),W(1)
C*****************TRANSMITTANCE FOR WATER VAPOR**********************
      IF (W(1).LT.1.0E-20) GO TO 500
      IF (I.LE.22) I1=I
      IF (I.GE.60.AND.I.LE.126) I1=I-37
      IF(I.GE.210.AND.I.LE.419) I1=I-120
      IF(I.GE.454.AND.I.LE.599) I1=I-154
      IF (I.GE.630) I1=I-184
      WS1=ALOG10(W(1))+C1(I1)
      TX(1)=EXP(-10**(-1.14619+0.55013*WS1))
  500 RETURN
      END
```

```
      SUBROUTINE H2O410(I,IV,W,C5,TX,SUM5)
C***WATER VAPOR CONTINUUM, 3-5 AND 8-12 MICRON REGIONS
      DIMENSION C5(6),TX(5),W(16)
      V=FLOAT(IV)
      IF(I.GT.109) GO TO 100
      IF(I.GT.22) GO TO 200
C***1.LE.I.LE.22:CALCULATE OPTICAL DEPTH DUE TO 8-12 MICRON CONTINUUM
      TX(5)=(4.18+5578.0*EXP(-7.87E-03*V))*W(5)
      GO TO 300
  200 CONTINUE
      IF(I.LT.63) GO TO 100
      IF(I.GT.68) GO TO 400
C***63.LE.I.LE.68:CALCULATE OPTICAL DEPTH DUE TO 4.6-4.8 MICRON
C***CONTINUUM.MODEL FROM BEN-SHALOM ET AL.1980,SPIE,VOL.253,261.
      II=I-62
      TX(5)=C5(II)*577.6*W(16)
      GO TO 300
  400 CONTINUE
      IF(I.LT.77) GO TO 100
C***77.LE.I.LE.109:CALCULATE OPTICAL DEPTH DUE TO 3.3-4.3 MICRON
C***CONTINUUM. MODEL FROM WATKINS ET AL.1979,APPL.OPT.,VOL.18,1149.
      V=V*1.0E-03
      V2=V*V
      V3=V2*V
      CBURCH=46.4745-48.0898*V+16.3988*V2-1.83217*V3
      CASL=-370.082+508.137*V-225.822*V2+32.7744*V3
      TX(5)=CBURCH*W(10)+CASL*W(15)
      GO TO 300
  100 TX(5)=0.0
  300 SUM5=TX(5)
      IF(TX(5).LT.1.0E-05) GO TO 500
      IF(TX(5).GT.20.0) GO TO 600
      TX(5)=EXP(-TX(5))
      RETURN
  500 TX(5)=1.0
      RETURN
  600 TX(5)=0.0
      RETURN
      END
```

```
      SUBROUTINE LTPATH(WLAY,WPATH,TBBY,ANGLE,LEN,ITYPE,H1,H2,MODEL)
      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL
      DIMENSION WLAY(34,16),TBBY(68),WPATH(68,16)
      COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1
      COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /SPOTLO/ISPOT,LOREAD,N16
      LOGICAL ISPOT,N16,LOREAD
555   IL=0
      IF (ITYPE.EQ.1) GO TO 1000
      IF (ITYPE.EQ.2.AND.H1.EQ.H2) J2=J1
      IF (H2.GT.H1.AND.ANGLE.GT.90..AND.NP1.EQ.1) J1=J1-1
      IF (JEXTRA.EQ.1) J2=J2+1
      IF ((ITYPE.EQ.2).AND.(H1.GT.H2).AND.(LENTOR.EQ.1)) J2=
     1     J2-1
      IF (ITYPE.EQ.3) J2=NL
      IF (.NOT.ISPOT) WRITE (IOOUT,1200)
      DO 100 IK=1,68
      TBBY(IK)=0.
      DO 100 K=1,KMAX
      WPATH(IK,K)=0.
  100 CONTINUE
      LEN=0
      NLL=NL-1
      IL=J1+1
      IJ=IL+NLL
      DO 200 K=1,KMAX
      E(K)=0.
  200 CONTINUE
      IF (ANGLE.GT.90.0) GO TO 300
      LEN=1.
      IL=J1-1
      HMIN=1.0E-6
      IJ=NLL
  300 CONTINUE
      DO 800 IK=1,68
      IF (LEN.EQ.0) IL=IL-1
      IF (LEN.EQ.1) IL=IL+1
      IJ=IJ-1
      IF (IL.EQ.0) GO TO 800
      DO 400 K=1,KMAX
      W(K)=E(K)+WLAY(IL,K)
      WPATH(IK,K)=W(K)
  400 CONTINUE
      IF (IL.LE.0.OR.IL.GE.NL) GO TO 500
      TBAR=(T(IL)+T(IL+1))*0.5
C     JEXTRA IS 1 ONLY WHEN PROGRAM NEVER LEAVES ONE LAYER
      IF (JEXTRA.EQ.1) TBAR=(T(J1)+T(J1+1))*0.5
  500 CONTINUE
      TBBY(IK)=TBAR
      DO 600 K=1,KMAX
      E(K)=W(K)
  600 CONTINUE
      IF (ANGLE.LE.90.0.AND.IL.EQ.NLL) GO TO 900
      IF (ITYPE.EQ.3.AND.ANGLE.LE.90.0) GO TO 700
      IF (ITYPE.EQ.3.AND.LEN.EQ.1.AND.IL.EQ.J2) GO TO 900
      IF (ITYPE.EQ.2.AND.LENTOR.EQ.0.AND.IL.EQ.J2) GO TO 900
      IF (IL.EQ.JMIN.AND.HMIN.GT.0) LEN=1
      IF (IL.EQ.1.AND.HMIN.LE.0.0) GO TO 900
      IF (LEN.EQ.0) GO TO 700
      IF (IL.EQ.JMIN.AND.IJ.EQ.IL+NLL) IL=IL-1
      IF (ITYPE.EQ.2.AND.IL.EQ.J2) GO TO 900
  700 CONTINUE
      IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=
     1     1,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK)
  800 CONTINUE
      IKMAX=68
      LEN=LENTOR
      RETURN
  900 CONTINUE
```

```
      IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=                 LTP00710
     1     1,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK)                  LTP00720
      IKMAX=IK                                                              LTP00730
      LEN=LENTOR                                                            LTP00740
      RETURN                                                                LTP00750
 1000 DO 1100 K=1,KMAX                                                      LTP00760
      WPATH(1,K)=W(K)                                                       LTP00770
 1100 CONTINUE                                                              LTP00780
      IF (MODEL.EQ.0) J1=1                                                  LTP00790
      J2=J1                                                                 LTP00800
      TBBY(1)=T(J1)                                                         LTP00810
      IKMAX=1                                                               LTP00820
      IK=1                                                                  LTP00830
      IF (.NOT.ISPOT) WRITE (IOOUT,1200)                                    LTP00840
      IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=                 LTP00850
     1     1,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK)                  LTP00860
      HMIN=1.0E-6                                                           LTP00870
      RETURN                                                                LTP00880
C                                                                           LTP00890
 1200 FORMAT (//,20X,37H CUMULATIVE ABSORBER AMOUNTS FOR THE ,              LTP00900
     1          16HATMOSPHERIC PATH,// 8X,3HH2O,5X,4HCO2+,6X,               LTP00910
     2          2HO3,7X,2HN2,6X,5HH2O C,4X,5HMOL S,4X,5HO3 UV,              LTP00920
     3          4X,5HH2O C,5X,4HHNO3,6X,3HSO2,6X,3HNH3,6X,3HNO2,            LTP00930
     4          5X,4HTAVE)                                                  LTP00940
 1300 FORMAT (I5,12E9.3,F10.3)                                              LTP00950
      END                                                                   LTP00960
```

```
      SUBROUTINE MOLSCT(IV,W,TX,SUM6)
C*********TRANSMITTANCE FOR MOLECULAR SCATTERING *********************
C    C6 EXPRESSION MODIFIED AS PER SHETTLE ET AL 1980,APPL.OPT.,VOL.19,
C    2873.
      DIMENSION TX(6),W(6)
      V=FLOAT(IV)
      C6=(V**4)/(9.67578E+18-1.11836E+09*V**2)
      TX(6)=C6*W(6)
      SUM6=TX(6)
      IF (TX(6).EQ.0.0) GO TO 200
      IF (TX(6).LE.0.1) GO TO 100
      IF (TX(6).GT.20.) GO TO 300
      TX(6)=EXP(-TX(6))
      GO TO 400
  100 TX(6)=1.0-TX(6)+0.5*TX(6)*TX(6)
      GO TO 400
  200 TX(6)=1.0
      GO TO 400
  300 TX(6)=0.0
  400 RETURN
      END
```

```
      SUBROUTINE NH3(I,W,C13,TX)
      COMMON /MO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),
     1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2
      DIMENSION C13(21),TX(13),W(13)
C***********************************************************************
C
C     THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY NH3 ( PPM READ IN
C     THE MAIN PROGRAM).
C
C***********************************************************************
      I1=I
      IF (W(13).LT.1.0E-20) GO TO 3
      WS13=ALOG10(W(13))+C13(I1)
      DO 1 J=1,9
      IF (WS13-FNH3(J)) 2,2,1
1     CONTINUE
2     TX(13)=EXP(-10**(FH1(J)+FH2(J)*WS13))
3     RETURN
      END
```

```
      SUBROUTINE NITRIC(I,W,C11,SUM11,TX)
      DIMENSION C11(4),TX(11),W(11)
C******** TRANSMITTANCE FOR NITRIC ACID********************************
      HABS=0.
      IF (I.LT.2.OR.I.GT.46) GO TO 100
      IF (I.GT.5.AND.I.LT.23) GO TO 100
      I1=I-1
      HABS=C11(I1)
  100 CONTINUE
      TX(11)=HABS*W(11)
      SUM11=TX(11)
      IF (TX(11).EQ.0.0) GO TO 300
      IF (TX(11).LE.0.1) GO TO 200
      IF (TX(11).GT.20.) GO TO 400
      TX(11)=EXP(-TX(11))
      GO TO 500
  200 TX(11)=1.0-TX(11)+0.5*TX(11)*TX(11)
      GO TO 500
  300 TX(11)=1.0
      GO TO 500
  400 TX(11)=0.0
  500 RETURN
      END
```

```
      SUBROUTINE NITRO(I,W,C4,TX,SUM4)
      DIMENSION C4(33),TX(4),W(4)
C*********TRANSMITTANCE FOR NITROGEN CONTINUUM**************************
      IF (I.LT.64) GO TO 200
      I1=I-63
C         TEMP FIX FOLLOWS
C     IF (I1.GT.10) GO TO 300
      TX(4)=C4(I1)*W(4)
      SUM4=TX(4)
      IF (TX(4).EQ.0.0) GO TO 200
      IF (TX(4).LE.0.1) GO TO 100
      IF (TX(4).GT.20.) GO TO 300
      TX(4)=EXP(-TX(4))
      GO TO 400
  100 TX(4)=1.0-TX(4)+0.5*TX(4)*TX(4)
      GO TO 400
  200 TX(4)=1.0
      GO TO 400
  300 TX(4)=0.0
  400 RETURN
      END
```

```
      SUBROUTINE NO2(I,W,C14,TX)
      COMMON /MO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),
     1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2
      DIMENSION C14(6),TX(14),W(14)
C***********************************************************************
C
C     THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY NO2 ( PPM READ IN
C     THE MAIN PROGRAM).
C
C***********************************************************************
      IF (I.GE.102.AND.I.LE.104) I1=I-98
      IF (I.LE.3) I1=I
      IF (W(14).LT.1.0E-20) GO TO 3
      WS14=ALOG10(W(14))+C14(I1)
      DO 1 J=1,9
      IF (WS14-FNO2(J)) 2,2,1
1     CONTINUE
2     TX(14)=EXP(-10**(O1(J)+O2(J)*WS14))
3     RETURN
      END
```

```
      SUBROUTINE OZONE(I,W,C3,TX)
      DIMENSION C3(86),TX(3),W(3)
C******************TRANSMITTANCE FOR OZONE*****************************
      IF (W(3).LT.1.0E-20) GO TO 500
      IF (I.LE.22) I1=I
      IF (I.GE.60) I1=I-37
      WS3=ALOG10(W(3))+C3(I1)
      TX(3)=1/(1+EXP(-3.08019+2.11127*WS3))
  500 RETURN
      END
```

```
      SUBROUTINE POINT (X,N,NP,TX)
      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL
      COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /SPOTLO/ISPOT,LOREAD,N16
      DIMENSION TX(16)
      LOGICAL ISPOT,N16,LOREAD
C***********************************************************************
C     SUBROUTINE POINT INTERPOLATES EXPONENTIALLY TO
C     DETERMINE THE EQUIVALENT ABSORBER AMOUNTS AT THAT ALTITUDE.
C
C       X IS THE HEIGHT IN QUESTION
C       N IS THE LEVEL INTEGER CORRESPONDING TO X OR THE LEVEL BELOW X
C       NP =1 IF X COINCIDES WITH MODEL ATMOSPHERE LEVEL ,IF NOT NP = 0
C       TX(1-8) ARE ABSORBER  AMOUNTS PER KM AT HEIGHT X
C***********************************************************************
555   N=NL
      NP=0
      IF (X.LT.0.0) X=0.
      IF (X.GT.Z(NL)) GO TO 400
      DO 100 I=1,NL
      N=I
      IF (X-Z(I)) 200,400,100
  100 CONTINUE
  200 J2=N
      N=N-1
      FAC=(X-Z(N))/(Z(J2)-Z(N))
      DO 300 K=1,KMAX
      IF (K.EQ.9.OR.K.EQ.7) GO TO 300
      TX(K)=0.0
      IF (EH(K,N).EQ.0.0) GO TO 300
      IF (EH(K,N).GT.1000.0) GO TO 300
      TX(K)=EH(K,N)*(EH(K,J2)/EH(K,N))**FAC
  300 CONTINUE
      GO TO 700
  400 NP=1
      DO 500 K=1,KMAX
  500 TX(K)=EH(K,N)
700   RETURN
      END
```

```
      FUNCTION RESFN (NR,WAVE)
C     THIS FUNCTION WILL READ IN UP TO 20 VALUES OF A RESPONSE FUNCTION
C     IF THE RESF CARD IS READ IN EOMAIN.  ONLY ONE RESPONSE FUNCTION
C     PER RUN IS ALLOWED.  THIS FUNCTION WILL ALSO DO A LINEAR INTERPOLA
C     OVER WAVELENGTH. IF RESF CARD IS NOT READ A VALUE OF 1 IS RETURNED
C     TO THE CALLING PROGRAM WHEN THIS FUNCTION IS REFERENCED.
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DIMENSION WAVELN(20),RESPFN(20)
      DATA WAVELN,RESPFN,ICOUNT,NBR /20*0.,20*0.,0,1/
      IF (NR.NE.1) GO TO 6
      ICOUNT=ICOUNT+1
      IF (ICOUNT.GT.1) GO TO 2
      READ (IOIN,100) NBR
      IF (NBR.GT.20) WRITE (IOOUT,102)
      IF (NBR.GT.20) STOP
      WRITE (IOOUT,103)
      DO 3 I=1,NBR
      READ (IOIN,101) WAVELN(I),RESPFN(I)
3     WRITE (IOOUT,104) WAVELN(I),RESPFN(I)
2     IF (WAVE.LT.(WAVELN(1)-.0001).OR.WAVE.GT.(WAVELN(NBR)+.0001))
     + GO TO 6
      DO 4 I=1,NBR
      K=I
4     IF (WAVE.GE.WAVELN(I)) GO TO 5
5     IF (WAVE/WAVELN(K).GE..99.AND.WAVE/WAVELN(K).LE.1.01) GO TO 7
      IF (K.EQ.NBR) GO TO 8
      RESFN=(WAVE-WAVELN(K))*(RESPFN(K+1)-RESPFN(K))/
     1 (RESPFN(K+1)-RESPFN(K))+RESPFN(K)
      RETURN
7     RESFN=RESPFN(K)
      RETURN
8     RESFN=RESPFN(NBR)
      RETURN
6     RESFN=1.
      RETURN
 100   FORMAT (I2)
 101   FORMAT (2(E10.4,1X))
102   FORMAT (1H ,51HTHE NUMBER OF VALUES FOR THE RESPONSE FUNCTION IS G
     +,58HREATER THAN THE DIMENSIONS LIMITS OF WAVELN( ) AND RESPFN( ),
     +/,1X,19HPROGRAM TERMINATED.)
103   FORMAT (1H ,20X,23HINPUT RESPONSE FUNCTION,/,1X,15X,10HWAVELENGTH,
     +5X,10HR FUNCTION)
104   FORMAT (1H ,15X,2(E10.4,1X))
      END
```

```
      SUBROUTINE SO2(I,W,C12,TX)
      COMMON /MO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),
     1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2
      DIMENSION C12(15),TX(12),W(12)
C*****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY SO2 ( PPM READ IN
C     THE MAIN PROGRAM).
C
C*****************************************************************
      IF (W(12).LT.1.0E-20) GO TO 5
      IF (I.GE.13.AND.I.LE.22) I1=I-12
      IF (I.GE.82) I1=I-71
      WS12=ALOG10(W(12))+C12(I1)
      DO 1 J=1,9
      IF(WS12-FS(J)) 2,2,1
1     CONTINUE
2     TX(12)=EXP(-10**(S1(J)+S2(J)*WS12))
5     RETURN
      END
```

```
      SUBROUTINE UNIMIX(I,W,C2,TX)
      DIMENSION C2(258),TX(2),W(2)
C*****************TRANSMITTANCE FOR UNIFORMLY MIXED GASES************
      IF (W(2).LT.1.0E-20) GO TO 500
      IF (I.LE.22) I1=I
      IF (I.GE.60.AND.I.LE.126) I1=I-37
      IF(I.GE.210.AND.I.LE.363) I1=I-120
      IF (I.GE.607) I1=I-363
      WS2=ALOG10(W(2))+C2(I1)
      TX(2)=EXP(-10**(-1.14619+0.55013*WS2))
  500 RETURN
      END
```

```
      SUBROUTINE UVOZNE(I,W,C8,TX,SUM8)
      DIMENSION C8(82),TX(8),W(8)
C******** TRANSMITTANCE FOR UV OZONE ***********************************
      AI=I
      IF(I.LE.1159) GO TO 90
      IF (I.GE.1335) GO TO 100
   90 XX=10.0
      XI=(AI-610.0)/XX+1.0
      L1=1
      L2=53
      GO TO 200
  100 XX=25.0
      XI=(AI-1335.0)/XX+57.0
      L1=57
      L2=102
  200 DO 300 N=L1,L2
      XD=XI-FLOAT(N)
      IF (XD) 500,400,300
  300 CONTINUE
  400 TX(8)=W(8)*C8(N)
      GO TO 600
  500 TX(8)=C8(N)+XD*(C8(N)-C8(N-1))
      TX(8)=W(8)*TX(8)
  600 SUM8=TX(8)
      IF (TX(8).EQ.0.0) GO TO 800
      IF (TX(8).LE.0.1) GO TO 700
      IF (TX(8).GT.20.0) GO TO 900
      TX(8)=EXP(-TX(8))
      GO TO 1000
  700 TX(8)=1.0-TX(8)+0.5*TX(8)*TX(8)
      GO TO 1000
  800 TX(8)=1.0
      GO TO 1000
  900 TX(8)=0.0
 1000 RETURN
      END
```

```
      SUBROUTINE SPOT(WAVN1,WAVN2,VIS,NR,IERR,MULDV)                    SPOT0010
C***********************************************************************SPOT0020
C     INPUT:  EXCLUDING THE OPTIONAL RESPONSE FUNCTION CARDS,           SPOT0030
C     THERE IS A MAXIMUM OF 7 CARDS TO EXECUTE THIS MODULE.             SPOT0040
C     THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OF      SPOT0050
C     THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN.         SPOT0060
C     THE CARDS ARE INPUT WITH FORMAT (A4,6X,7E10.4)                    SPOT0070
C     EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4          SPOT0080
C     FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL PER           SPOT0090
C     FIELD BEGINNING IN COL 11.                                        SPOT0100
C             THE CARDS ARE NOT ORDER DEPENDENT.                        SPOT0110
C     CARD 1                                                            SPOT0120
C  ENVR       ISORC, ITARG, IHAZE, MODEL, NLAM                          SPOT0130
C             ISORC = 0 SUNLIGHT ONLY                                   SPOT0140
C                     1 MOONLIGHT ONLY                                  SPOT0150
C                     2 EMISSION ONLY                                   SPOT0160
C                     3 SUNLIGHT AND EMISSION                           SPOT0170
C                     4 MOONLIGHT AND EMISSION                          SPOT0180
C             ITARG = 0 BACKGROUND ONLY                                 SPOT0190
C                     1 GROUND REFLECTANCE/EMISSION                     SPOT0200
C                     2 TARGET REFLECTANCE/EMISSION                     SPOT0210
C  ** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK **SPOT0220
C     FOR SLANT PATHS IHAZE = 1,2, OR 3 ARE THE ONLY ALLOWED VALUES.    SPOT0230
C           IHAZE    =0,NO AEROSOL ATTENUATION                          SPOT0240
C                    =1, MARITIME POLAR                                 SPOT0250
C                    =2, MARITIME ARCTIC                                SPOT0260
C                    =3, CONTINENTAL POLAR                              SPOT0270
C                    =4, RAIN                                           SPOT0280
C                    =5, SNOW                                           SPOT0290
C                    =7, USER SUPPLIED EXTINCTION COEFFICIENT           SPOT0300
C                        (READ ON ATM CARD - SEE CARD 3 BELOW)          SPOT0310
C                    =8, EXTINCTION COEFFICIENT WILL BE READ FROM       SPOT0320
C                        PHASE FUNCTION DATA FILE                       SPOT0330
C             MODEL = 1 TROPICAL MODEL ATMOSPHERE                       SPOT0340
C                     2 MIDLATITUDE SUMMER                              SPOT0350
C                     3 MIDLATITUDE WINTER                              SPOT0360
C                     4 SUBARCTIC SUMMER                                SPOT0370
C                     5 SUBARCTIC WINTER                                SPOT0380
C                     6 1962 US STANDARD                                SPOT0390
C                     8 ISRAELI STANDARD (YEAR, DAYTIME)                SPOT0400
C                     9 ISRAELI STANDARD (YEAR, NIGHTTIME)              SPOT0410
C             NLAM    OPTION FOR AEROSOL PHASE FUNCTION                 SPOT0420
C                     =  0 NO AEROSOL ATTENUATION                       SPOT0430
C                     NE 0 READ PHASE FUNCTION DATA SET - ALSO SEE      SPOT0440
C                          ID BELOW AND EXPLN OF PFN DATA SET BELOW     SPOT0450
C             ID     PHASE FUNCTION IDENTIFIER                          SPOT0460
C                    =0,   USER SUPPLIED                                SPOT0470
C                    =1,   MARITIME ARCTIC, VIS=0.1 TO 2.0 KM           SPOT0480
C                    =2,   MARITIME POLAR, VIS=0.2 KM                   SPOT0490
C                    =3,   MARITIME POLAR, VIS=02. KM                   SPOT0500
C                    =4,   CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM        SPOT0510
C                    =5,   WHITE PHOSPHORUS                             SPOT0520
C                    =6,   HEXACHLOROETHANE                             SPOT0530
C                    =7,   FOG OIL                                      SPOT0540
C                    =8,   DUST (MODERATE AEROSOL LOADING)              SPOT0550
C                    =9,   DUST (HEAVY AEROSOL LOADING)                 SPOT0560
C                    =10, MARITIME MODEL B, VIS=5KM, RH=95%             SPOT0570
C                    =11, MARITIME MODEL B, VIS=10KM,RH=90%             SPOT0580
C                    =12, MARITIME MODEL B, VIS=50KM,RH=50%             SPOT0590
C     CARD 2    ****  IF ISORC LT 2 OR ITARG LT 1 THIS CARD IS NOT NEEDEDSPOT0600
C  EMIS    EM(1), TM(1), EM(2), TM(2)                                   SPOT0610
C           EM(1)       EMISSIVITY OF GROUND                            SPOT0620
C           TM(1)       TEMPERATURE OF GROUND (KELVIN)                  SPOT0630
C           EM(2)       EMISSIVITY OF TARGET                            SPOT0640
C           TM(2)       TEMPERATURE OF TARGET (KELVIN)                  SPOT0650
C     CARD 3                                                            SPOT0660
C  ATM     ZENTH, CLDHGT, PHASE,BETAEX                                  SPOT0670
C           ZENTH       INCIDENT ANGLE OF SUNLIGHT OR MOONLIGHT (DEGREES)SPOT0680
C           CLDHGT      HEIGHT OF BOTTOM OF CLOUD LAYER (KM)            SPOT0690
C                       ONLY NEEDED WHEN IHAZE NE 0  (DEFAULT IS 0.)    SPOT0700
```

```
C              PHASE       PHASE ANGLE FOR MOONLIGHT (DEGREES)
C              BETAEX      USER SUPPLIED EXTINCTION COEFFICIENT
C                          VALID ONLY WHEN IHAZE=7
C         CARD 4
C     TARG    RTARG,COSX,COSY,COSZ
C              RTARG       SLANT RANGE FROM RECEIVER TO TARGET (KM)
C              COSX        X-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)
C              COSY        Y-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)
C              COSZ        Z-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)
C         CARD 5   ****  IF ITARG LT 1 THIS CARD IS NOT NEEDED
C     REFL    A0(1),A1(1),IALB(1),A0(2),A1(2),IALB(2)
C              A0(1)       ALBEDO COEFFICIENT FOR GROUND
C              A1(1)       ALBEDO COEFFICIENT FOR GROUND
C              IALB(1)     TYPE OF REFLECTION DISTRIBUTION FOR GROUND
C              A0(2)       ALBEDO COEFFICIENT FOR TARGET
C              A1(2)       ALBEDO COEFFICIENT FOR TARGET
C              IALB(2)     TYPE OF REFLECTION DISTRIBUTION FOR TARGET
C              IALB   = 0 LAMBERTIAN REFLECTION SURFACE
C                       1 ISOTROPIC
C         CARD 6
C     SENS    ALT, THETA, PHI, SANG2
C              ALT         ALTITUDE OF RECEIVER (KM)
C              THETA       POLAR DIRECTION OF LOOK ANGLE (DEGREES)
C              PHI         AZMITH DIRECTION OF LOOK ANGLE (DEGREES)
C
C*** METEOROLOGICAL AZIMUTH CONVENTION ASSUMED: N = 0 DEG,
C*** E = 90 DEG, S = 180 DEG, W = 270 DEG ************************
C
C              SANG2       HALF ANGLE DEFINING RECEIVER FIELD-OF-VIEW
C                            (DEGREES)
C         CARD 7
C         GO           SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FOR
C                      THIS CALL.  NOTE THAT IF A DATA CARD IS NOT READ
C                      THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLS
C                      TO THE MODULE ARE STILL IN EFFECT.
C
C     OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY NR=1 IN EOMAIN)
C     THESE CARDS MUST FOLLOW THE GO CARD AND CAN ONLY BE INSERTED ONCE
C         CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2).
C         CARDS 2 - NUMBER OF VALUES: FORMAT (2(E10.4,1X))
C                     ONE VALUE OF WAVE (UM) AND RESPONSE FUNCTON PER CARD
C                       N.B. ONLY ONE RESPONSE FUNCTION PER EOSAEL RUN.
C
C       AUXILLARY READ FROM UNIT IPHFUN (ASL DATA SET PROVIDED WITH EOSAEL)
C              ANG         ANGLES AT WHICH PHASE MATRIX IS DEFINED,
C                          NANG VALUES  (DEFAULT IS 65), FORMAT(11(F6.2,1X))
C              NANG,ID,WAVE,OMEGA0,BETAEX,BETABS
C                          NUMBER OF ANGLES AT WHICH THE PHASE FUNCTION HAS
C                          VALUES, PFN IDENTIFIER, WAVELENGTH(UM), SINGLE
C                          SCATTERING ALBEDO, EXTINCTION COEFFICIENTS (TOTAL
C                          AND SCATTERING).
C                          FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X)).
C              PF          PHASE FUNCTION AT SPECIFIED ANGLES,
C                          FORMAT (6(E12.6,1X))
C*********************************************************************
      LOGICAL L1,L2,L3,L4,L5,L6,L7,ISPOT,N16,LOREAD,HORIZ
      DIMENSION DUMMY(16)
      EQUIVALENCE (ITARG,IT)
      COMMON /ANSW2/TTR(16),TBR(16),CNTRST(16)
      COMMON /ALBED/A0(2),A1(2),IALB(2)
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
      COMMON /CGEOM/COSGM,COSBT,COSIN
      COMMON /COMI1/ISORC,ITARG,IWN,JHL
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /CTARG/RTARG,COSX,COSY,COSZ
      COMMON /EMISS/EM(2),TM(2)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /M01/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,
     +            RRS(16,34),SCOE(16,34),
     +              TRANS(16,5),RADA(16,2),WAVE(16),SS(16),
```

```
     1        DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),
     2        PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)
      COMMON /MO2/ WO(34),RO,TBOUND,JP,IM,ML,IP,JSTOR
      COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL
      COMMON /SPOTLO/ ISPOT,LOREAD,N16
      COMMON /LOWEX/ WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX,
     1 CLDHGT,NCLD
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),
     1 BE(16),SINGWV,PF(65),LMAX
      COMMON /LOGIC/L1,L2,L3,L4,L5,L6,L7
      DATA L1,L2,L3,L4,L5,L6,L7/7*.FALSE./
      DATA ITR1,ITR2,ITR3,ITR4/2,3,5,1/
      N16=.TRUE.
      INBR=16
      DUM=1.
      ICLMAT=0
      LOREAD=.TRUE.
      ISPOT=.TRUE.
C     INITIALIZE AND READ INPUT PARAMETERS
      CALL ZERO
      CALL INDAT(IEMISS,IHAZE,IM,LEN,ML,MODEL,
     1     TBOUND,0,CLDHGT,BETAEX)
C     CHECK FOR ERROR IN INPUT DATA
      IF (IHAZE.EQ.9) IERR=1
      IF (IERR.EQ.1) RETURN
      IF (ISORC.NE.2) ISWTCH=1
C     FIRST CALL IS TO READ LOWTRAN DATA FILE ONLY
      CALL LT4M(ALT,DUM,ZENTH,3,0,TRANS(1,1),DUMMY,DUMMY,
     1                  IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,
     2     ICLMAT,IERR,NR,IHAZE,MULDV)
      V1=WAVN1
      V2=WAVN2
      CALL CKER(V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,DUM)
      WAVE(1)=10000./V1
      DO 300 IW=2,INBR
      V2=V1+20.*FLOAT(MULDV)*FLOAT(IW-1)
      IF (V2.GE.WAVN2) GO TO 400
      WAVE(IW)=10000./V2
  300 CONTINUE
      L1=.TRUE.
C   MAXIMUM NO. OF WAVELENGTHS
  400 IWN=IW-1
      IF (L1) IWN=INBR
      NWVL=IWN
C
C     ARRAY WVL IS USED ONLY IN SUBROUTINE PFUNC. THE WAVELENGTHS
C     IN THIS ARRAY INCREASE WITH INCREASING ARRAY INDEX. THE
C     VALID RANGES FOR VALUES IN THE WVL ARRAY ARE : 0.2-2.0, 3.0-5.0,
C     AND 8.0-12.0 MICROMETERS.
C
      DO 355 JX=1,NWVL
      INDM=NWVL-JX+1
  355 WVL(JX)=WAVE(INDM)
      DO 500 I=1,NL
      IF (ALT.LE.Z(I)) GO TO 600
  500 CONTINUE
      WRITE (IOOUT,3700) ALT,I,Z(I)
      IERR=1
      RETURN
  600 CONTINUE
      IF (I.EQ.1) WRITE (IOOUT,3800) ALT
      IF (I.EQ.1) IERR=1
      IF (IERR.EQ.1) RETURN
      JHL=I-1
      NLL=NL-1
      SANG=TWOPI*(1.0-COS(SANG2*PIRAD))
      CZNTH=COS(ZENTH*PIRAD)
      SZNTH=SIN(ZENTH*PIRAD)
      CTHTA=COS(THETA*PIRAD)
      STHTA=SIN(THETA*PIRAD)
```

```
      CPHI=COS(PHI*PIRAD)
      SPHI=SIN(PHI*PIRAD)
      IF (ISORC.EQ.2) GO TO 900
C     CALCULATE SOURCE TERM ... FOR SUNLIGHT (ISORC=0,3)
C                               OR MOONLIGHT (ISORC=1,4)
      DO 700 IW=1,IWN
      SS(IW)=0.0
      IF (ISORC.EQ.0.OR.ISORC.EQ.3) SS(IW)=SOLARS(WAVE(IW))
      IF (ISORC.EQ.1.OR.ISORC.EQ.4) SS(IW)=SMOON(WAVE(IW),PHASE)
  700 CONTINUE
C     CALCULATE DIRECT INTENSITY ... FOR SUNLIGHT (ISORC=0,3)
C                                    OR MOONLIGHT (ISORC=1,4)
      COSIN=STHTA*CPHI*SZNTH+CTHTA*CZNTH
      ANGIN=ACOS(COSIN)/PIRAD
C     READ PHASE FUNCTION FILE
      CALL INDAT(IEMISS,IHAZE,IM,LEN,ML,MODEL,
     1     TBOUND,1,CLDHGT,BETAEX)
      IF (ANGIN.GT.SANG2) L2=.TRUE.
      IF (ZENTH.GT.80.0) L3=.TRUE.
      CALL LT4M(ALT,DUM,ZENTH,3,0,TRANS(1,1),DUMMY,DUMMY,
     1                  IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,
     2     ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (L2.OR.L3) GO TO 900
      DO 800 IW=1,IWN
      DIR(IW)=SS(IW)*TRANS(IW,1)
  800 CONTINUE
  900 IF (ITARG.EQ.0.AND.THETA.GT.90.0) L7=.TRUE.
      IF (L7) GO TO 3500
      IF (ITARG.EQ.0) GO TO 1200
      IF (ITARG.EQ.1) GO TO 1000
C     TARGET...
      ZTARG=RTARG*CTHTA+ALT
      COSTX=STHTA*CPHI
      COSTY=STHTA*SPHI
      COSTZ=CTHTA
      COSBT=COSX*SZNTH+COSZ*CZNTH
      COSGM=-(COSTX*COSX+COSTY*COSY+COSTZ*COSZ)
      IF (THETA.LE.90.0) GO TO 1100
      COSBTG=CZNTH
      COSGMG=COS((180.0-THETA)*PIRAD)
      GO TO 1100
C     GROUND...
 1000 ZTARG=0.0
      COSBT=CZNTH
      COSGM=COS((180.0-THETA)*PIRAD)
      COSBTG=COSBT
      COSGMG=COSGM
C     CALCULATE ATMOSPHERIC TRANSMISSION/RADIANCE FOR VARIOUS PATHS
 1100 IF (COSGM.LE.0.0) L4=.TRUE.
      IF (COSBT.LE.0.0) L5=.TRUE.
      IF (THETA.LE.90.0.AND.ITARG.EQ.1) L6=.TRUE.
      IF (L4.OR.L6) GO TO 3500
 1200 CONTINUE
      IF (THETA.EQ.90.0)
     1     CALL LT4M(ALT,DUM,1000.0,1,2,TRANS(1,4),RADA(1,1),
     2    DUMMY,            IEMISS,LEN,MODEL,VIS,    V1,V2,
     3    TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (THETA.EQ.90.0.AND.ITARG.EQ.2)
     1     CALL LT4M(ALT,DUM,RTARG,1,2,TRANS(1,2),RADA(1,2),
     2    DUMMY,            IEMISS,LEN,MODEL,VIS,    V1,V2,
     3    TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (THETA.EQ.90.0) GO TO 1300
      IF (THETA.LT.90.0)
     1     CALL LT4M(ALT,DUM,THETA,3,2,TRANS(1,4),RADA(1,2),DUMMY,
     2               IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,
     3    ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (ITARG.EQ.0) GO TO 2000
      IF (THETA.LT.90.0)
     1     CALL LT4M(ALT,ZTARG,THETA,2,2,TRANS(1,2),RADA(1,2)
     2    ,DUMMY,             IEMISS,LEN,MODEL,VIS,    V1,V2,
```

```
     3    TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (THETA.LT.90.0) GO TO 1300
      CALL LT4M(ALT,ZTARG,THETA,2,2,TRANS(1,2),RADA(1,2),
     1      RADG(1),               IEMISS,LEN,MODEL,VIS,    V1,V2,
     2      TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (ITARG.NE.2) ITR4=2
      IF (ITARG.NE.2) GO TO 1300
      IF (ZTARG.LE.0.0) ITR4=2
      IF (ZTARG.LE.0.0) GO TO 1300
      ITR1=4
      CALL LT4M(ALT,0.0,THETA,2,2,TRANS(1,4),RADA(1,1),
     1      RADG(1),               IEMISS,LEN,MODEL,VIS,    V1,V2,
     2      TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
 1300 IF (ISORC.LT.2) GO TO 1600
C     CALCULATE UNCOLLIDED EMISSION ... FROM GROUND (ITARG=1)
C                                         OR TARGET (ITARG=2)
      DO 1500 IW=1,IWN
      WAVEM=WAVE(IW)/1.0E+4
      IF (ITARG.EQ.2) GO TO 1400
      BKG(IW)=BLACK(WAVEM,TGRD)*EM(1)
      RADG(IW)=RADG(IW)*EM(1)*COSGMG
      BK(IW)=BKG(IW)
      UTEM(IW)=RADG(IW)
      GO TO 1500
 1400 BK(IW)=BLACK(WAVEM,TM(2))*EM(2)
      UTEM(IW)=BK(IW)*COSGM*TRANS(IW,2)
      IF (THETA.LE.90.0) GO TO 1500
      BKG(IW)=BLACK(WAVEM,TGRD)*EM(1)
      RADG(IW)=RADG(IW)*EM(1)*COSGMG
 1500 CONTINUE
 1600 IF (ISORC.EQ.2) GO TO 2900
C     CALCULATE UNCOLLIDED REFLECTANCE ... FROM GROUND (ITARG=1)
C                                            OR TARGET (ITARG=2)
      IEMISS=0
      IF (L5) GO TO 2000
      HORIZ=ABS(ZTARG-ALT).LT.0.001
      IF(HORIZ) ITR2=1
      IF(HORIZ) GO TO 1700
      CALL LT4M(ZTARG,DUM,ZENTH,3,2,TRANS(1,3),DUMMY,DUMMY,
     1                   IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,
     2      ICLMAT,IERR,NR,IHAZE,MULDV)
 1700 IF (ZTARG.LE.0.0) ITR3=ITR2
      IF (ZTARG.LE.0.0) GO TO 1800
      IF (THETA.LE.90.0) GO TO 1800
      CALL LT4M(0.0,DUM,ZENTH,3,2,TRANS(1,5),DUMMY,DUMMY,
     1                   IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,
     2      ICLMAT,IERR,NR,IHAZE,MULDV)
 1800 DO 1900 IW=1,IWN
      ALB=ALBEDO(IT)
      UTRF(IW)=SS(IW)*COSBT*ALB*TRANS(IW,2)*TRANS(IW,ITR2)
      IF (THETA.LE.90.0) GO TO 1900
      UERF(IW)=SS(IW)*COSBTG*ALBEDO(1)*TRANS(IW,ITR1)*
     1        TRANS(IW,ITR3)
 1900 CONTINUE
C     CALCULATE SINGLE-SCATTERED PATH RADIANCE ...
C                FROM SUNLIGHT (ISORC=0,3)
C                 OR MOONLIGHT (ISORC=1,4)
 2000 CALL COEFS(P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX)
      IF (IERR.EQ.1) RETURN
      IF (ITARG.EQ.0) GO TO 2700
      CALL PATHRD(CTHTA,ALT,RTARG,1,IHAZE,NR,
     1      IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)
      DO 2100 IW=1,IWN
 2100 PATHR2(IW)=SS(IW)*PATHR(IW)
      IF (ABS(CTHTA).LE.1.0E-3) GO TO 2600
      IF (CTHTA.LT.0.0) GO TO 2300
 2200 Z2=Z(NLL)
      IO=2
      RT=(Z(NLL)-ZTARG)/CTHTA
      GO TO 2800
```

```
 2300 IF (ZTARG.GT.0.0) GO TO 2500
      DO 2400 IW=1,IWN
 2400 PATHR(IW)=0.0
      GO TO 2900
 2500 Z2=0.0
      IO=1
      RT=-ZTARG/CTHTA
      GO TO 2800
 2600 Z2=ZTARG
      RT=1000.0
      IO=3
      GO TO 2800
 2700 IEMISS=0
      ZTARG=ALT
      IF (ABS(CTHTA).LE.1.0E-03) GO TO 2600
      GO TO 2200
 2800 CALL PATHRD(CTHTA,ZTARG,RT,IO,IHAZE,NR,
     1      IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)
C     CALCULATE BACKGROUND AND TOTAL INTENSITY, PLUS
C               CONTRAST RATIO
 2900 DO 3100 IW=1,IWN
      PATHR(IW)=PATHR2(IW)+SS(IW)*PATHR(IW)
      TTR(IW)=PATHR2(IW)+UTRF(IW)+RADA(IW,2)+UTEM(IW)
      TBR(IW)=PATHR(IW)+UERF(IW)+RADA(IW,ITR4)+RADG(IW)
      DIF=TTR(IW)-TBR(IW)
      IF (TBR(IW).GT.0.0) GO TO 3000
      IF (TTR(IW).EQ.TBR(IW)) CNTRST(IW)=0.0
      IF (TTR(IW).NE.TBR(IW)) CNTRST(IW)=1.0E30
      GO TO 3100
 3000 CNTRST(IW)=DIF/TBR(IW)
 3100 CONTINUE
C     CALCULATE TOTAL RADIANCES INTEGRATED OVER DETECTOR RESPONSE
      DV2=DV*0.5
      SUMRPF=0.
      DO 3200 IW=1,IWN
      NW=10000./WAVE(IW)
      W2=10000./(FLOAT(NW)-DV2)
      W1=10000./(FLOAT(NW)+DV2)
      IF (IW.EQ.1) W2=WAVE(1)
      IF (IW.EQ.IWN) W1=WAVE(IWN)
      RESPFN=RESFN(NR,WAVE(IW))
      SUMRPF=SUMRPF+RESPFN
      DW=(W2-W1)*RESPFN
      TOT(1)=TOT(1)+DW*UTEM(IW)
      TOT(2)=TOT(2)+DW*UTRF(IW)
      TOT(3)=TOT(3)+DW*RADA(IW,2)
      TOT(4)=TOT(4)+DW*PATHR2(IW)
      TOT(5)=TOT(5)+DW*TTR(IW)
      TOT(6)=TOT(6)+DW*RADG(IW)
      TOT(7)=TOT(7)+DW*UERF(IW)
      TOT(8)=TOT(8)+DW*RADA(IW,ITR4)
      TOT(9)=TOT(9)+DW*PATHR(IW)
      TOT(10)=TOT(10)+DW*TBR(IW)
 3200 TOT(11)=TOT(11)+DW*DIR(IW)
      IF (NR.NE.1) SUMRPF=1.
      DO 3250 I=1,11
3250  TOT(I)=TOT(I)/SUMRPF
      IF (TOT(10).GT.0.0) GO TO 3300
      IF (TOT(5).EQ.TOT(10)) TOT(12)=0.0
      IF (TOT(5).NE.TOT(10)) TOT(12)=1.0E30
      GO TO 3400
 3300 TOT(12)=(TOT(5)-TOT(10))/TOT(10)
 3400 CONTINUE
 3500 CALL OUTPUT(MODEL,IHAZE,CLDHGT)
      RETURN
C
 3700 FORMAT (1H ,11H ALTITUDE (,F10.3,17H) GREATER THAN Z(,
     1        I2,2H)=,F10.3,27H  CONTROL RETURNED TO MAIN
     2        10HFROM SPOT.)
 3800 FORMAT (1H ,11H ALTITUDE (,F10.3,16H) LESS THAN ZERO,
```

```
1 37H  CONTROL RETURNED TO MAIN FROM SPOT.)                         SPOT4210
  END                                                               SPOT4220
```

```
      FUNCTION ALBEDO(I)
      COMMON /ALBED/A0(2),A1(2),IALB(2)
      COMMON /CGEOM/COSGM,COSBT,COSIN
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C     CALCULATE ALBEDO FOR GROUND (ITARG=1) OR TARGET (ITARG=2)
C     IALB(I) = 0    LAMBERTIAN REFLECTION SURFACE
C               1    ISOTROPIC
      A=A0(I)+A1(I)*COSBT
      IF (IALB(I).EQ.0) ALBEDO=A*COSGM/PI
      IF (IALB(I).EQ.1) ALBEDO=A/TWOPI
      RETURN
      END
```

```
      FUNCTION BLACK(W,T)
C     BLACK(W,T) = PLANCK FUNCTION (UNITS: WATT PER SQUARE METER PER
C     MICROMETER PER STERADIAN), GIVEN WAVELENGTH W IN CM AND TEMP-
C     ERATURE T IN K
C
C     EXP OVERFLOW PROTECTION
C
      ARG=1.43879/(W*T)
      IF(ARG.LT.88.) GO TO 1
      BLACK=0.0
      RETURN
    1 BLACK=1.19106E-12/(W**5*(EXP(ARG)-1.0))
      RETURN
      END
```

```
      SUBROUTINE COEFS(P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX)             COE00010
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB                 COE00020
      COMMON /CTARG/RTARG,COSX,COSY,COSZ                                COE00030
      COMMON /MO1/DUMMIE(715),MHOLD,NL,                                 COE00040
     +                 RRS(16,34),SCOE(16,34),                          COE00050
     +                 TRANS(16,5),RADA(16,2),WAVE(16),SS(16),          COE00060
     1        DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),                COE00070
     2        PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)             COE00080
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),     COE00090
     1 BE(16),SINGWV,PF(65)                                             COE00100
      COMMON /COMI1/ISORC,ITARG,IWN,JHL                                 COE00110
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                    COE00120
      DIMENSION P(34),T(34)                                             COE00130
C     CALCULATE THE WAVELENGTH-DEPENDENT CONSTANT PRESSURE COEFFICIENTS COE00140
C     FOR MOLECULAR SCATTERING.                                         COE00150
C   LOOP OVER LAYERS                                                    COE00160
      DO 600 I=1,NL                                                     COE00170
      PS=P(I)/1013.0                                                    COE00180
      TS=CDEGK/T(I)                                                     COE00190
      RSCAT=PS*TS                                                       COE00200
C   LOOP OVER WAVELENGTHS                                               COE00210
      DO 600 IW=1,IWN                                                   COE00220
      RAYS=0.0                                                          COE00230
      NW=10000./WAVE(IW)                                                COE00240
C     RAYLEIGH SCATTERING = 0.FOR WAVELENGTH GT 3.33 UM                 COE00250
      IF (NW.LT.3000) GO TO 200                                         COE00260
      WN=FLOAT(NW)                                                      COE00270
      RAYS=RSCAT*(WN**4)/(9.67578E+18-1.11836E+09*WN**2)                COE00280
200   CONTINUE                                                          COE00290
      AEXT=1.                                                           COE00300
      AABS=1.                                                           COE00310
      IF (IHAZE.EQ.0.OR.I.NE.(NCLD-1)) GO TO 1                          COE00320
      EXT55=3.912/VIS                                                   COE00330
C     UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE            COE00340
      ZTALT=ZTARG/ALT                                                   COE00350
      IF (ABS(ZTALT-1.).LT..01) RNG=RTARG                               COE00360
      IF ((ZTARG.GT.ALT).AND.(RTARG.GT..5/COS(THETA*PIRAD)))            COE00370
     1  RNG=.5/COS(THETA*PIRAD)                                         COE00380
      IF (ZTARG.LT.RTARG.AND.(RTARG.GT..5/COS((180.-THETA)*PIRAD)))     COE00390
     1        RNG=.5/COS((180.-THETA)*PIRAD)                            COE00400
      IF (ITARG.EQ.0.AND.(RTARG.GT..5/COS(THETA*PIRAD)))                COE00410
     1        RNG=.5/COS(THETA*PIRAD)                                   COE00420
      ISLANT=1                                                          COE00430
      IF (ABS(ZTALT-1.).LT..01) ISLANT=0                                COE00440
C     CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL        COE00450
      CALL XSCALE(WAVE(IW),88.,EXT55,XSTRN,IERR,ISLANT,IHAZE,RNG,THETA) COE00460
      IF (IERR.EQ.1) RETURN                                             COE00470
      AEXT=-ALOG(XSTRN)/RNG                                             COE00480
C     USER SUPPLIED COEFF(IHAZE=7), OR READ FROM PFN DATA FILE(IHAZE=8) COE00490
      IF (IHAZE.EQ.7) AEXT=BETAEX                                       COE00500
      IF (IHAZE.EQ.8) AEXT=BE(IW)                                       COE00510
      IF (WAVE(IW).LT.2.) AABS=1.                                       COE00520
      IF (WAVE(IW).GE.3..AND.WAVE(IW).LE.5.) AABS=AEXT*.2               COE00530
      IF (WAVE(IW).GE.8..AND.WAVE(IW).LE.12.) AABS=AEXT*.45             COE00540
1     CONTINUE                                                          COE00550
      SCOE(IW,I)=AEXT-AABS+RAYS                                         COE00560
C     CHECK FOR NO AEROSOL PRESENT                                      COE00570
      IF(SCOE(IW,I).LT.1.E-20)RRS(IW,I)=1.0                             COE00580
C     AEROSOL AND RAYLEIGH PRESENT                                      COE00590
      IF(SCOE(IW,I).GE.1.E-20)RRS(IW,I)=RAYS/SCOE(IW,I)                 COE00600
C     CHECK FOR NO RAYLEIGH SCATTERING                                  COE00610
      IF (RAYS.LT.1.E-20) RRS(IW,I)=0.0                                 COE00620
  600 CONTINUE                                                          COE00630
      RETURN                                                            COE00640
      END                                                               COE00650
```

```
      SUBROUTINE DIAG
C        THIS SUBROUTINE PRODUCES DIAGNOSTIC COMMENTARY FOR THE
C        SPOT PROGRAM.
C
C        CALLING SEQUENCE:  CALL DIAG
C
C        EXTERNAL VARIABLES REQUIRED:
C                 THETA (COMMON BLOCK BKDAT)
C                 HLIGHT,HTYPE (COMMON BLOCK HOLRTH)
C                 L1,L2,L3,L4,L5,L6 (COMMON BLOCK LOGIC)
C
      LOGICAL L1,L2,L3,L4,L5,L6,L7
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
      COMMON /HOLRTH/                  HITARG(8,3),HISORC(6,5),
     1       HMODEL(5,6),HLIGHT(3),HMNLT(3),HSNLT(3),
     2       HTRGT(2),HTYPE(2),HGRND(2)
      COMMON /LOGIC/L1,L2,L3,L4,L5,L6,L7
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DATA I/1/
C        WRITE HEADING.
      WRITE (IOOUT,900)
C        IF ERRORS, GO TO 5; OTHERWISE PRINT CLEAN RUN MESSAGE.
      IF (L1.OR.L2.OR.L3.OR.L4.OR.L5.OR.L6.OR.L7) GO TO 100
      WRITE (IOOUT,1000)
      GO TO 700
C        HERE IF THERE WERE ERRORS
  100 WRITE (IOOUT,1100)
      IF (.NOT.L1) GO TO 200
      WRITE (IOOUT,1200) I
      I=I+1
  200 IF (.NOT.L2) GO TO 300
      WRITE (IOOUT,1300) I,HLIGHT
      I=I+1
  300 IF (.NOT.L3) GO TO 400
      WRITE (IOOUT,1400) I,HLIGHT
      I=I+1
  400 IF (.NOT.L4) GO TO 500
      WRITE (IOOUT,1500) I,HTYPE
      I=I+1
  500 IF (.NOT.L5) GO TO 600
      WRITE (IOOUT,1600) I,HLIGHT
      I=I+1
  600 IF (.NOT.L6) GO TO 700
      WRITE (IOOUT,1700) I,THETA
      I=I+1
  700 IF (.NOT.L7) GO TO 800
      WRITE (IOOUT,1800) I,THETA
C        WRITE FOOTING.
  800 WRITE (IOOUT,1900)
C
  900 FORMAT (1H0,21X,90(1H*),3(/,21X,1H*,88X,1H*))
 1000 FORMAT (21X,1H*,28X,29HNO SPOT DIAGNOSTICS FOR THIS
     1          3HRUN,28X,1H*,/,21X,1H*,28X,8H-- ---- ,11(1H-)
     2          ,13H --- ---- ---,28X,1H*)
 1100 FORMAT (21X,1H*,28X,25HSPOT DIAGNOSTIC MESSAGES
     1          7HFOLLOW:,28X,1H*,/,21X,1H*,28X,5H---- ,10(1H-
     2          ),1X,8(1H-),1X,6(1H-),29X,1H*,2(/,21X,1H*,88X,
     3          1H*))
 1200 FORMAT (21X,1H*,9X,I1,30H.  NUMBER OF WAVELENGTHS (IWN)
     1          9H EXCEEDS ,31HALLOWABLE DIMENSIONS; IWN RESET
     2          ,8X,1H*,/,21X,1H*,13X,6HTO 16.,69X,1H*,/,21X,
     3          1H*,88X,1H*)
 1300 FORMAT (21X,1H*,9X,I1,12H.  NO DIRECT,3A4,9HINCIDENT
     1          7HWITHIN ,25HRECEIVER'S FIELD OF VIEW.,13X,1H*
     2          ,/,21X,1H*,88X,1H*)
 1400 FORMAT (21X,1H*,9X,I1,25H.  ANGLE OF INCIDENCE FOR,
     1          3A4,8HGREATER ,21HTHAN 80.0 DEGREES; NO,12X,
     2          1H*,/,21X,1H*,13X,26HCALCULATIONS WILL BE MADE
     3          14HFOR ITARG = 0.,35X,1H*,/,21X,1H*,88X,1H*)
 1500 FORMAT (21X,1H*,9X,I1,1H.,1X,2A4,14HDOES NOT FACE
```

```
     1          9HRECEIVER.,45X,1H*,/,21X,1H*,88X,1H*)                        DIAG0710
1600 FORMAT (21X,1H*,9X,I1,1H.,3A4,22HILLUMINATES BACK SIDE                   DIAG0720
     1          3HOF ,7HTARGET.,33X,1H*,/,21X,1H*,88X,1H*)                    DIAG0730
1700 FORMAT (21X,1H*,9X,I1,30H.  THETA LESS THAN 90 DEGREES                   DIAG0740
     1          10HAND ITARG 13H = 1; THETA =,F6.4,19X,1H*,/                  DIAG0750
     2          ,21X,1H*,88X,1H*)                                             DIAG0760
1800 FORMAT (21X,1H*,9X,I1,25H.  THETA GREATER THAN 90                        DIAG0770
     1          13HDEGREES AND  19HITARG = 0; THETA = ,F7.4,                  DIAG0780
     2          13X,1H*,/,21X,1H*,88X,1H*)                                    DIAG0790
1900 FORMAT (2(21X,1H*,88X,1H*,/),21X,90(1H*))                                DIAG0800
     RETURN                                                                   DIAG0810
     END                                                                      DIAG0820
```

```
      SUBROUTINE INDAT(IEMISS,IHAZE,IM,LEN,ML,
     1           MODEL,TBOUND,ISW,CLDHGT,BETAEX)
      COMMON /ALBED/A0(2),A1(2),IALB(2)
      COMMON /MO1/DUMMIE(715),MHOLD,NLHOLD,DUMMYS(1088),
     +             TRANS(16,5),RADA(16,2),WAVE(16),SS(16),
     1        DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),
     2        PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
      COMMON /COMI1/ISORC,ITARG,IWN,JHL
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      COMMON /CTARG/RTARG,COSX,COSY,COSZ
      COMMON /EMISS/EM(2),TM(2)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON /GEOMET/PTS(15),IGEOSW
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),
     1 BE(16),SINGWV,PF(65),LMAX
      DIMENSION DAT(7),TYPE(7)
      DATA TYPE /4HENVR,4HEMIS,4HATM ,4HTARG,4HREFL,4HSENS,4HGO  /
      DATA IZERO /0/
C
C     SUBROUTINE INDAT IS CALLED UPON TO A) READ INPUT CONTROL
C     PARAMETERS, WITH CARD ORDER INDEPENDENT INPUT (SEE SPOT
C     FOR MORE DETAIL) AND, B) TO READ VALUES OF THE PHASE
C     FUNCTION AT SPECIFIED ID AND WAVELENGTH.
C     NOTE.  IF ISORC LT 2 OR ITARG LT 1 EM(1),TM(1),EM(2),TM(2)
C            ARE NOT NEEDED.
C            IF ITARG LT 1 A0(),A1(),IALB() ARE NOT NEEDED.
C
C     ISW=1 ON SECOND CALL TO INDAT
      IF (ISW.EQ.1) GO TO 400
      IF (IZERO.GT.0) GO TO 9
      ISORC=0
      ITARG=0
      MODEL=0
      IHAZE=0
      NLAM=0
      EM(1)=0.
      TM(1)=0.
      EM(2)=0.
      TM(2)=0.
      ZENTH=0.
      CLDHGT=0.
      PHASE=0.
      RTARG=0.
      COSX=0.
      COSY=0.
      COSZ=0.
      A0(1)=0.
      A1(1)=0.
      IALB(1)=0
      A0(2)=0.
      A1(2)=0.
      IALB(2)=0
      ALT=0.
      THETA=0.
      PHI=0.
      SANG2=0.
      IZERO=1
9     CONTINUE
      WRITE(IOOUT,600)
  600 FORMAT(1H0,'SPOT CONTROL CARDS READ FOR THIS RUN:'/)
      DO 10 I=1,7
      READ (IOIN,11) (DAT(J),J=1,7)
11    FORMAT (A4,6X,7E10.4)
      WRITE(IOOUT,610)(DAT(J),J=1,7)
  610 FORMAT(1H ,A4,6X,7E10.4)
      IF (DAT(1).EQ.TYPE(1)) GO TO 1
      IF (DAT(1).EQ.TYPE(2)) GO TO 2
      IF (DAT(1).EQ.TYPE(3)) GO TO 3
      IF (DAT(1).EQ.TYPE(4)) GO TO 4
```

```
      IF (DAT(1).EQ.TYPE(5)) GO TO 5
      IF (DAT(1).EQ.TYPE(6)) GO TO 6
      IF (DAT(1).EQ.TYPE(7)) GO TO 7
C     ERROR RETURN
      WRITE (IOOUT,101)
101   FORMAT(1H ,48HINCORRECT INPUT CARD FOR SPOT, CONTROL RETURNED ,
     1 18HTO MAIN FROM INDAT)
      IHAZE=9
      RETURN
C     OPERATING ENVIRONMENT
1     ISORC=IFIX(DAT(2))
      ITARG=IFIX(DAT(3))
      IHAZE=IFIX(DAT(4))
      MODEL=IFIX(DAT(5))
      NLAM=IFIX(DAT(6))
      ID=IFIX(DAT(7))
      GO TO 8
C     EMISSIVITY AND TEMPERATURE OF GROUND AND TARGET, RESPECTIVELY
2     EM(1)=DAT(2)
      TM(1)=DAT(3)
      EM(2)=DAT(4)
      TM(2)=DAT(5)
      TBOUND=TM(1)
      GO TO 8
C     INCIDENT ZENITH ANGLE OF RADIATION, CLOUD BOTTOM HEIGHT,
C     PHASE ANGLE OF MOON, OPTIONAL EXTN COEF (VALID WHEN IHAZE=8)
3     ZENTH=DAT(2)
      CLDHGT=DAT(3)
      PHASE=DAT(4)
      BETAEX=DAT(5)
      GO TO 8
C     TARGET PROPERTIES
4     RTARG=DAT(2)
      COSX=DAT(3)
      COSY=DAT(4)
      COSZ=DAT(5)
      GO TO 8
C     ALBEDO COEFFICIENTS AND TYPE OF REFLECTION SURFACE FOR
C     GROUND AND TARGET, RESPECTIVELY
5     A0(1)=DAT(2)
      A1(1)=DAT(3)
      IALB(1)=IFIX(DAT(4))
      A0(2)=DAT(5)
      A1(2)=DAT(6)
      IALB(2)=IFIX(DAT(7))
      GO TO 8
C     SENSOR CHARACTERISTICS
6     ALT=DAT(2)
      THETA=DAT(3)
      PHI=DAT(4)
C
C    EXPECTING INPUT AZIMUTH IN METEOROLOGICAL CONVENTION
C    (I.E., N = 0 DEG, E = 90 DEG, S = 180 DEG, W = 270 DEG),
C    SO CONVERT TO MATHEMATICAL CONVENTION FOR PURPOSES OF
C    SPOT (ASSUMING Y-AXIS IS POSITIVE NORTHWARD, X-AXIS POSI-
C    TIVE EASTWARD).
C
      PHI=90.-PHI
      SANG2=DAT(5)
8     CONTINUE
10    CONTINUE
7     CONTINUE
      IEMISS=0
      IM=0
      LEN=0
      ML=0
      IF (ISORC.GT.1) IEMISS=1
      COSX=COS(COSX*PIRAD)
      COSY=COS(COSY*PIRAD)
      COSZ=COS(COSZ*PIRAD)
```

```
C      GEOMETRICAL OPTION
       IF(IGEOSW.NE.1)GO TO 311
       RTARG=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+
      +(PTS(6)-PTS(3))**2)
       THETA=ACOS((PTS(3)-PTS(6))/RTARG)
       RTDCON=1.0/PIRAD
       THETA=THETA*RTDCON
       ALT=PTS(6)
       DELX=PTS(1)-PTS(4)
       DELY=PTS(2)-PTS(5)
       HDIS=SQRT(DELX**2+DELY**2)
       PHI=ACOS(DELX/HDIS)
       PHI=PHI*RTDCON
       IF(DELY.LT.0.0)PHI=360.0-PHI
  311  CONTINUE
       IF (ITARG.EQ.1) RTARG=ALT/ABS(COS(THETA*PIRAD))
       IF (ITARG.EQ.0) RTARG=1000.0
       RETURN
  400  CONTINUE
       REWIND IPHFUN
       DO 500 I=1,IWN
500    PF(I)=0.
       IF (NLAM.NE.0) CALL PFUNC(ID)
       RETURN
       END
```

```
      SUBROUTINE OUTPUT(MODEL,IHAZE,CLDHGT)
      LOGICAL LMNLT
      COMMON /MO1/ DUMMIE(715),MHOLD,NLHOLD,DUMMYS(1088),
     +               TRANS(16,5),RADA(16,2),WAVE(16),SS(16),
     1         DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),
     2         PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
      COMMON /COMI1/ISORC,ITARG,IWN,JHL
      COMMON /ANSW2/TTR(16),TBR(16),CNTRST(16)
      COMMON /HOLRTH/                HITARG(8,3),HISORC(6,5),
     1         HMODEL(5,6),HLIGHT(3),HMNLT(3),HSNLT(3),
     2         HTRGT(2),HTYPE(2),HGRND(2)
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DATA HGRND/4H GRO,4HUND /
      DATA HISORC/4HSUNL,4HIGHT,4H ONL,4HY   ,4H    ,4H    ,
     1      4HMOON,4HLIGH,4HT ON,4HLY  ,4H    ,4H    ,4HEMIS,
     2      4HSION,4H ONL,4HY   ,4H    ,4H    ,4HSUNL,4HIGHT,
     3      4H AND,4H EMI,4HSSIO,4HN   ,4HMOON,4HLIGH,4HT AN,
     4      4HD EM,4HISSI,4HON  /
      DATA HITARG/4HND R,4HEFLE,4HCTAN,4HCE  ,4H    ,4H    ,
     1      4H    ,4H    ,4HGROU,4HND R,4HEFLE,4HCTAN,4HCE /,
     2      4H EMI,4HSSIO,4HN   ,4HTARG,4HET R,4HEFLE,4HCTAN,
     3      4HCE /,4H EMI,4HSSIO,4HN   /
      DATA HMODEL/4HTROP,4HICAL,4H    ,4H    ,4H    ,4HMIDA,
     1      4HLTIT,4HUDE ,4HSUMM,4HER  ,4HMIDA,4HLTIT,4HUDE ,
     2      4HWINT,4HER  ,4HSUBA,4HRCTI,4HC SU,4HMMER,4H    ,
     3      4HSUBA,4HRCTI,4HC WI,4HNTER,4H    ,4H1962,4H U.S,
     4      4H. ST,4HANDA,4HRD  /
      DATA HMNLT/4H  MO,4HONLI,4HGHT /
      DATA HSNLT/4H  SU,4HNLIG,4HHT  /
      DATA HTRGT/4H TAR,4HGET /
      DATA HTYPE/4H    ,4H    /
      DATA LMNLT/.FALSE./
      IF (ISORC.NE.0.AND.ISORC.NE.3) GO TO 200
      DO 100 I=1,3
  100 HLIGHT(I)=HSNLT(I)
      GO TO 400
  200 DO 300 I=1,3
  300 HLIGHT(I)=HMNLT(I)
      LMNLT=.TRUE.
  400 IF (ITARG.EQ.0) GO TO 600
      IF (ITARG.EQ.2) GO TO 500
      HTYPE(1)=HGRND(1)
      HTYPE(2)=HGRND(2)
      GO TO 600
  500 HTYPE(1)=HTRGT(1)
      HTYPE(2)=HTRGT(2)
  600 CALL DIAG
      IF (MODEL.GT.7) GO TO 700
      WRITE (IOOUT,1400) ISORC,(HISORC(I,ISORC+1),I=1,6),
     1      ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,
     2      (HMODEL(I,MODEL),I=1,5),IHAZE
      GO TO 750
  700 IF(MODEL.EQ.8)WRITE (IOOUT,1450) ISORC,(HISORC(I,ISORC+1),I=1,6),
     1      ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE
      IF(MODEL.EQ.9)WRITE (IOOUT,1500) ISORC,(HISORC(I,ISORC+1),I=1,6),
     1      ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE
  750 IF (IHAZE.GT.0) WRITE (IOOUT,1600) CLDHGT
  800 WRITE (IOOUT,1700)
      IF (LMNLT) WRITE (IOOUT,1800) PHASE
      WRITE (IOOUT,1900) HLIGHT
      DO 900 I=1,IWN
      NW=10000./WAVE(I)
  900 WRITE (IOOUT,2000) WAVE(I),NW,SS(I),BK(I),BKG(I)
      WRITE (IOOUT,2100)
      DO 1000 I=1,IWN
      NW=10000./WAVE(I)
 1000 WRITE (IOOUT,2200) WAVE(I),NW,UTEM(I),UTRF(I),
     1      RADA(I,2),PATHR2(I),TTR(I)
      WRITE (IOOUT,2300)
```

```
      DO 1100 I=1,IWN
      NW=10000./WAVE(I)
 1100 WRITE (IOOUT,2200) WAVE(I),NW,RADG(I),UERF(I),
     1        RADA(I,1),PATHR(I),TBR(I)
      WRITE (IOOUT,2400) HLIGHT
      IF (LMNLT) WRITE (IOOUT,1800) PHASE
      WRITE (IOOUT,2500) HLIGHT,HLIGHT
      DO 1200 I=1,IWN
      NW=10000./WAVE(I)
 1200 WRITE (IOOUT,2600) WAVE(I),NW,SS(I),DIR(I)
      WRITE (IOOUT,2700)
      DO 1300 I=1,IWN
      NW=10000./WAVE(I)
 1300 WRITE (IOOUT,2800) WAVE(I),NW,TTR(I),TBR(I),CNTRST(I)
      WRITE (IOOUT,2900) (TOT(I),I=1,10),TOT(12),HLIGHT,
     1        TOT(11)
      RETURN
C
 1400 FORMAT (43X,37HDEFINITION  OF  CONTROL  PARAMETERS
     1          8HFOLLOWS:,/,43X,10(1H-),6H  --   ,7(1H-),2X,
     2          10(1H-),2X,7(1H-),///,43X,9HPARAMETER,3X,
     3          5HVALUE,3X,11HDESCRIPTION,/,43X,9(1H-),3X,5(
     4          1H-),3X,11(1H-),//,45X,5HISORC,7X,I1,5X,6A4,//
     5          ,45X,5HITARG,7X,I1,5X,8A4,//,45X,5HMODEL,7X,
     6          I1,5X,5A4,//,45X,5HIHAZE,7X,I1,/)
 1450 FORMAT (43X,37HDEFINITION  OF  CONTROL  PARAMETERS
     1          8HFOLLOWS:,/,43X,10(1H-),6H  --   ,7(1H-),2X,
     2          10(1H-),2X,7(1H-),///,43X,9HPARAMETER,3X,
     3          5HVALUE,3X,11HDESCRIPTION,/,43X,9(1H-),3X,5(
     4          1H-),3X,11(1H-),//,45X,5HISORC,7X,I1,5X,6A4,//
     5          ,45X,5HITARG,7X,I1,5X,8A4,//,45X,5HMODEL,7X,
     6 I1,5X,32HISRAELI STANDARD (YEAR, DAYTIME)//45X,5HIHAZE,7X,I1/)
 1500 FORMAT (43X,37HDEFINITION  OF  CONTROL  PARAMETERS
     1          8HFOLLOWS:,/,43X,10(1H-),6H  --   ,7(1H-),2X,
     2          10(1H-),2X,7(1H-),///,43X,9HPARAMETER,3X,
     3          5HVALUE,3X,11HDESCRIPTION,/,43X,9(1H-),3X,5(
     4          1H-),3X,11(1H-),//,45X,5HISORC,7X,I1,5X,6A4,//
     5          ,45X,5HITARG,7X,I1,5X,8A4,//,45X,5HMODEL,7X,
     6 I1,5X,34HISRAELI STANDARD (YEAR, NIGHTTIME)//45X,5HIHAZE,7X,I1/)
1600  FORMAT (1H ,45X,22HCLOUD BOTTOM HEIGHT = ,F5.3,3H KM)
 1700 FORMAT (1H1,56X,19HSOURCE  INTENSITIES,/,56X,7H------
     1          12H -----------,//)
 1800 FORMAT (44X,27HPHASE ANGLE FOR MOONLIGHT: ,F6.2,
     1          10H (DEGREES),//)
 1900 FORMAT (15X,10HWAVELENGTH,3X,10HWAVENUMBER,3X,3A4,
     1          6HSOURCE,11X,13HTARGET SOURCE,15X,7HGROUND
     2          6HSOURCE,/,15X,9H(MICRONS),6X,6H(CM-1),11X,
     3          8HSTRENGTH,17X,8HSTRENGTH,20X,8HSTRENGTH,/
     4          ,41X,20H(WATTS M-2 MICRON-1),2(3X,9H(WATTS M-
     5          16H2 MICRON-1 SR-1)),/,15X,10(1H-),3X,10(1H-),
     6          3X,20(1H-),3X,25(1H-),3X,25(1H-),//)
 2000 FORMAT (15X,1PE10.4,3X,I7,11X,1PE10.4,15X,1PE10.4,18X,
     1          1PE10.4)
 2100 FORMAT (1H1,46X,33HCOMPONENTS  FOR  RADIANCE  FROM
     1          6HTARGET,/,46X,29H----------  ---  --------  --
     2          10H--  ------,//,53X,23H(WATTS M-2 MICRON-1 SR-
     3          2H1),///,22X,10HWAVELENGTH,3X,10HWAVENUMBER,
     4          5X,^HTARGET,7X,6HTARGET,7X,7HPARTIAL,6X,
     5          7HPARTIAL,6X,5HTOTAL,/,22X,9H(MICRONS),6X,
     6          6H(CM-1),6X,8HEMISSION,4X,11HREFLECTANCE,2X,
     7          11HATMOSPHERIC,5X,4HPATH,8X,6HTARGET,/,75X,
     8          8HEMISSION,5X,8HRADIANCE,5X,8HRADIANCE,/
     9          ,22X,10(1H-),3X,10(1H-),4X,8(1H-),4X,11(1H-),
     X          2X,11(1H-),3X,8(1H-),5X,8(1H-),//)
 2200 FORMAT (22X,1PE10.4,3X,I7,3X,1P5E13.4)
 2300 FORMAT (1H1,47X,29HCOMPONENTS  FOR  BACKGROUND
     1          8HRADIANCE,/,47X,27H----------  ---  ----------
     2          10H  --------,//,53X,23H(WATTS M-2 MICRON-1 SR-
     3          2H1),///,22X,10HWAVELENGTH,3X,10HWAVENUMBER,
     4          5X,6HGROUND,7X,6HGROUND,8X,5HTOTAL,8X,5HTOTAL,
```

```
     5          7X,5HTOTAL,/,22X,9H(MICRONS),6X,6H(CM-1),6X,                     OUTP1410
     6          8HEMISSION,4X,11HREFLECTANCE,2X,11HATMOSPHERIC                   OUTP1420
     7          ,5X,4HPATH,6X,10HBACKGROUND,/,75X,8HEMISSION,                    OUTP1430
     8          2(5X,8HRADIANCE),/,22X,10(1H-),3X,10(1H-),4X,                    OUTP1440
     9          8(1H-),4X,11(1H-),2X,11(1H-),3X,8(1H-),4X,10(                    OUTP1450
     X          1H-),//)                                                         OUTP1460
 2400 FORMAT (1H1,58X,6HDIRECT,3A4,/,57X,6(1H-),1X,12(1H-),/                     OUTP1470
     1          /,56X,20H(WATTS M-2 MICRON-1),//)                                OUTP1480
 2500 FORMAT (41X,10HWAVELENGTH,3X,10HWAVENUMBER,2X,3A4,1X,                      OUTP1490
     1          3A4,/,41X,9H(MICRONS),6X,6H(CM-1),7X,6HSOURCE,                   OUTP1500
     2          8X,4HFLUX,/,68X,8HSTRENGTH,/,41X,10(1H-),3X,                     OUTP1510
     3          10(1H-),2X,12(1H-),1X,12(1H-),//)                                OUTP1520
 2600 FORMAT (41X,1PE10.4,3X,I7,6X,1PE10.4,3X,1PE10.4)                           OUTP1530
 2700 FORMAT (1H1,58X,15HTOTAL  RADIANCE,/,58X,11H-----  ----                    OUTP1540
     1          4H----,//,53X,25H(WATTS M-2 MICRON-1 SR-1),///                   OUTP1550
     2          ,35X,10HWAVELENGTH,3X,10HWAVENUMBER,5X,                          OUTP1560
     3          6HTARGET,5X,10HBACKGROUND,4X,8HCONTRAST,/                        OUTP1570
     4          ,35X,9H(MICRONS),6X,6H(CM-1),33X,5HRATIO,/                       OUTP1580
     5          ,35X,10(1H-),3X,10(1H-),5X,6(1H-),5X,10(1H-),                    OUTP1590
     6          4X,8(1H-),//)                                                    OUTP1600
 2800 FORMAT (35X,1PE10.4,3X,I7,3X,1P3E13.4)                                     OUTP1610
 2900 FORMAT (1H1,46X,30HDETECTOR-RESPONSE  WAVELENGTH-                          OUTP1620
     1          10HINTEGRATED,/,46X,17(1H-),2X,21(1H-),//                        OUTP1630
     2          ,58X,16H(WATTS M-2 SR-1),//,46X,7HTARGET                         OUTP1640
     3          8HEMISSION,16X,1PE10.4,/,46X,7HTARGET                            OUTP1650
     4          11HREFLECTANCE,13X,1PE10.4,/,46X,8HPARTIAL                       OUTP1660
     5          20HATMOSPHERIC EMISSION,3X,1PE10.4,/,46X,                        OUTP1670
     6          21HPARTIAL PATH RADIANCE,10X,1PE10.4,/,46X,                      OUTP1680
     7          21HTOTAL TARGET RADIANCE,10X,1PE10.4,/,46X,                      OUTP1690
     8          15HGROUND EMISSION,16X,1PE10.4,/,46X,7HGROUND                    OUTP1700
     9          11HREFLECTANCE,13X,1PE10.4,/,46X,6HTOTAL                         OUTP1710
     X          20HATMOSPHERIC EMISSION,5X,1PE10.4,/,46X,                        OUTP1720
     1          19HTOTAL PATH RADIANCE,12X,1PE10.4,/,46X,                        OUTP1730
     2          25HTOTAL BACKGROUND RADIANCE,6X,1PE10.4,//                       OUTP1740
     3          ,46X,41(1H*),//,46X,8HCONTRAST,22X,1PE11.4,//                    OUTP1750
     4          ,46X,41(1H*),//,46X,6HDIRECT,3A4,13X,1PE10.4,/                   OUTP1760
     5          ,46X,11H(WATTS M-2))                                             OUTP1770
      END                                                                        OUTP1780
```

```
      SUBROUTINE PATHRD(CTHED,HP,RT,IO,IHAZE,NR,
     1             IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)
      LOGICAL HORIZ
      DIMENSION TR1(16),TR2(16),DUMMY(16),ANS(16)
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),
     1 BE(16),SINGWV,SAER(65),LDUM
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
      COMMON /CGEOM/COSGM,COSBT,CSA
      COMMON /COMI1/ISORC,ITARG,IWN,JHL
      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,
     +              RRS(16,34),SCOE(16,34),
     +              TRANS(16,5),RADA(16,2),WAVE(16),SS(16),
     1        DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),
     2        PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)
C **** CONST1 = (1.-.0295)*.5*3./(8.*PI)/(1.+.5*.0295)
C **** CONST2 = .0295/(1.+.5*.0295)*3./(8.*PI)
      DATA CONST1,CONST2/.0570805145,.0034701189/
      NLL=NL-1
      SRAYL=CONST1*(1.+CSA*CSA)+CONST2
C **** INITIALIZE VARIABLES ****
      DO 800 IW=1,IWN
  800 PATHR(IW)=0.0
      DO 900 J=2,NLL
      IF (HP.LT.Z(J)) GO TO 1000
  900 CONTINUE
      J=NLL
 1000 JU=J
      JL=JU-1
      HORIZ=.FALSE.
      DS1=0.0
      H2=HP
      RAT=(H2-Z(JL))/(Z(JU)-Z(JL))
      GO TO (1100,1200,1300),IO
 1100 IX=RT*1.99999+1
      DX=RT/FLOAT(IX)
      DH=DX*CTHED
      HORIZ=ABS(CTHED).LT.1.E-3
      DIST=0.5*DX
      GO TO 1400
 1200 IX=NLL-JL
      DH=Z(JU)-HP
      DX=DH/CTHED
      GO TO 1400
C     HORIZONTAL PATH FOLLOWS
 1300 IX=100
      DIST=DIST-0.5*DX+0.25
      DX=0.5
      HORIZ=.TRUE.
      GO TO 1500
 1400 H2=HP+DH*0.5
 1500 DO 2600 K=1,IX
      IF (HORIZ) GO TO 2000
      CALL LT4M(HP,H2,THETA,2,2,TR1,DUMMY,DUMMY,
     1      IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,ICLMAT,IERR,
     2  NR,IHAZE,MULDV)
      CALL LT4M(H2,DUM,ZENTH,3,2,TR2,DUMMY,DUMMY,
     1      IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,ICLMAT,IERR,
     2  NR,IHAZE,MULDV)
      IF (IO.EQ.2) GO TO 1800
      DO 1600 J=2,NLL
      IF (H2.LT.Z(J)) GO TO 1700
 1600 CONTINUE
      J=NLL
 1700 JU=J
      JL=JU-1
      RAT=(H2-Z(JL))/(Z(JU)-Z(JL))
 1800 DO 1900 IW=1,IWN
      SC=SCOE(IW,JL)+RAT*(SCOE(IW,JU)-SCOE(IW,JL))
      RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL))
      SSCAT=RS*SRAYL+(1.0-RS)*SAER(IW)
```

```
1900  PATHR(IW)=PATHR(IW)+TR1(IW)*TR2(IW)*SC*SSCAT*DX
      GO TO (2400,2500,2300),IO
 2000 CONTINUE
      CALL LT4M(HP,DUM,DIST,1,2,TR1,DUMMY,DUMMY,
     1     IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,ICLMAT,IERR,
     2  NR,IHAZE,MULDV)
      IF (K.EQ.1) CALL LT4M(HP,DUM,ZENTH,3,2,TR2,DUMMY,DUMMY,
     1               IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,
     2     ICLMAT,IERR,
     2  NR,IHAZE,MULDV)
      DS=0.0
      DO 2200 IW=1,IWN
      IF (K.GT.1) GO TO 2100
      SC=SCOE(IW,JL)+RAT*(SCOE(IW,JU)-SCOE(IW,JL))
      RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL))
      ANS(IW)=TR2(IW)*SC*(RS*SRAYL+(1.0-RS)*SAER(IW))
 2100 DPATH=ANS(IW)*TR1(IW)*DX
      DS=DS+DPATH
 2200 PATHR(IW)=PATHR(IW)+DPATH
      DS1=DS1+DS
      DS=DS*0.5/DX
      IF (IO.EQ.3.AND.K.GT.1.AND.DS/DS1.LT.0.001) RETURN
      IK=(K/20)*20
      IF (IO.EQ.3.AND.IK.EQ.K) DX=DX*2.0
 2300 DIST=DIST+DX
      GO TO 2600
 2400 H2=H2+DH
      GO TO 2600
 2500 JU=JU+1
      JL=JL+1
      IF (K.EQ.1) RAT=0.5
      H2=(Z(JU)+Z(JL))*0.5
      DX=(Z(JU)-Z(JL))/CTHED
 2600 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ZERO                                                   ZER00010
      COMMON /M01/ DUMMIE(715),M,NL,DUMMYS(1088),RDATA(300)             ZER00020
      DO 100 I=1,300                                                    ZER00030
      RDATA(I)=0.0                                                      ZER00040
  100 CONTINUE                                                          ZER00050
      RETURN                                                            ZER00060
      END                                                               ZER00070
```

```
      SUBROUTINE CLIMAT(LOCAT,MONTH,NHOUR,IWIND,NPRT,TEMP,PRESS,RH,AH,DP
     1,VIS,WNDVEL,WINDIR,IPASCT)
C
C ********************************************************************
C
C               CLIMATOLOGY MODULE - CLIMAT
C
C     PURPOSE - TO PROVIDE THE CENTRAL EUROPEAN AND MID-EASTERN
C                 CLIMATOLOGY DATA REQUIRED BY OTHER MODULES OF EOSAEL.
C
C     PARAMETER DESCRIPTION
C          LOCAT - CLIMATOLOGY REGION INDICATOR. LOCAT IS AN INTEGER
C                    (1-4) FOR CENTRAL EUROPE AND
C                    (5-10) FOR MID-EAST.
C
C                 REGION  1 - EUROPEAN LOWLANDS,
C                 REGION  2 - EUROPEAN RHINE VALLEY,
C                 REGION  3 - EUROPEAN HIGHLANDS,
C                 REGION  4 - EUROPEAN ALPINE,
C                 REGION  5 - MIDEAST DESERTS,
C                 REGION  6 - MIDEAST COASTAL,
C                 REGION  7 - MIDEAST PERSIAN GULF,
C                 REGION  8 - MIDEAST RED SEA,
C                 REGION  9 - MIDEAST EASTERN MOUNTAINS, AND
C                 REGION 10 - MIDEAST INDUS VALLEY.
C
C          MONTH - AN INTEGER (1-12) INDICATING THE MONTH OF THE YEAR.
C                    MONTH IS USED TO SELECT THE SEASON WHICH IS
C                    APPLICABLE TO THE REGION LOCAT.
C          NHOUR - AN INTEGER (0-23) INDICATING THE TIME OF DAY LOCAL
C                    STANDARD TIME (LST). NHOUR IS USED TO SELECT ONE OF
C                    FOUR TIME PERIODS OF THE DAY 20-02, 03-09, 10-14,
C                    AND 15-19.
C          IWIND - *** NOT USED ***
C          NPRT - A PRINT SELECTOR.
C                   NPRT LE ZERO - DO NOT PRINT CLIMATOLOGICAL DATA.
C                   NPRT GT ZERO - PRINT ALL AVAILABLE MEANS, STANDARD
C                     DEVIATIONS, AND PERCENT OCCURRENCES.
C          TEMP - MEAN TEMPERATURE (C).
C          PRESS - MEAN SEA LEVEL PRESSURE (MB).
C          RH - MEAN RELATIVE HUMIDITY (PERCENT).
C          AH - MEAN ABSOLUTE HUMIDITY (GM/CU M).
C          DP - MEAN DEW-POINT TEMPERATURE (C).
C          VIS - MEAN HORIZONTAL VISIBILITY (KM).
C          WNDVEL - MEAN WIND SPEED (MPS).
C          WINDIR - MOST PROBABLE WIND DIRECTION (DEGREES). WINDIR IS
C                     GIVEN IN 30 DEGREE INCREMENTS (015,045,075,...
C                     ,345).
C          IPASCT - INDICATOR (1-6) FOR THE MOST PROBABLE PASQUILL
C                     STABILITY CATEGORY (A-F).
C          CLDHT - MEAN CLOUD HEIGHT (KM).
C          CLDCVR - MEAN TOTAL CLOUD COVER (PERCENT).
C          WNDDIR - WIND DIRECTION (DEGREES).
C
C     SUBROUTINES AND FUNCTIONS - NONE
C
C     CARD INPUT - NONE
C
C     TAPE INPUT - YES. BE SURE TO ASSIGN THE CLIMATOLOGY DATA TAPE TO
C                    UNIT NCLIMT.
C
C ********************************************************************
C
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      DIMENSION REGION(70),SEASON(8),HOUR(8),DATA(18),DIR(13)
C
      DATA REGION/4HEURO,4HPEAN,4H LOW,4HLAND,4HS    ,
     1 2*4H    ,4HEURO,4HPEAN,4H RHI,4HNE V,4HALLE,4HY    ,4H    ,4HEURO,
     2 4HPEAN,4H HIG,4HHLAN,4HDS   ,2*4H    ,4HEURO,4HPEAN,4H ALP,4HINE ,
     3 3*4H    ,4HMIDE,4HAST ,4HDESE,4HRTS ,3*4H    ,4HMIDE,4HAST ,4HCOA
```

```
     4S,4HTAL  ,3*4H    ,4HMIDE,4HAST ,4HPERS,4HIAN ,4HGULF,2*4H    ,4HMI
     5DE,4HAST ,4HRED  ,4HSEA  ,3*4H    ,4HMIDE,4HAST ,4HEAST,4HERN ,4HMOU
     6N,4HTAIN,4HS    ,4HMIDE,4HAST ,4HINDU,4HS VA,4HLLEY,2*4H    /
      DATA SEASON/4HWINT,4HER  ,4HSPRI,4HNG  ,4HSUMM,4HER  ,4HAUTU,4HMN
     1 /
      DATA HOUR/4H20-0,4H2   ,4H03-0,4H9   ,4H10-1,4H4   ,4H15-1,4H9   /
      DATA DIR/4H 015,4H 045,4H 075,4H 105,4H 135,4H 165,4H 195,4H 225,
     14H 255,4H 285,4H 315,4H 345,4H VBL/
C
C          POSITION THE TAPE NCLIMT FOR READING
C
      REWIND NCLIMT
C
C          SKIP OVER ALL DATA FOR REGIONS 1,2,...,LOCAT-1
C
      IF(LOCAT.LT.1.OR.LOCAT.GT.10) LOCAT=1
      LSKIP=1056*(LOCAT-1)
C
C          SKIP OVER DATA FOR SEASONS 1,2,...,SEASON-1 FOR REGION LOCAT
C
      NSEASN=1
      IF(MONTH.GE.3.AND.MONTH.LE.5) NSEASN=2
      IF(MONTH.GE.6.AND.MONTH.LE.8) NSEASN=3
      IF(MONTH.GE.9.AND.MONTH.LE.10) NSEASN=4
      IF(LOCAT.GE.5.AND.MONTH.EQ.11) NSEASN=4
      NSKIP=LSKIP+176*(NSEASN-1)
C
C          SKIP OVER DATA FOR TIME PERIODS 0,1,...,PERIOD-1 FOR REGION
C                LOCAT DURING SEASON.
C
      NTIME=1
      IF(NHOUR.GE.3.AND.NHOUR.LE.9) NTIME=2
      IF(NHOUR.GE.10.AND.NHOUR.LE.14) NTIME=3
      IF(NHOUR.GE.15.AND.NHOUR.LE.19) NTIME=4
      NSKIP=NSKIP+44*(NTIME-1)
      IF(NSKIP.LE.0) GO TO 2
      DO 1 J=1,NSKIP
      READ(NCLIMT,9) A
    1 CONTINUE
C
C          IF NPRT GT 0, PRINT A HEADING FOR THE THERMODYNAMIC DATA
C
    2 CONTINUE
      IF(NPRT.LE.0) GO TO 3
      WRITE(IOOUT,15)
      ILOC=7*LOCAT-6
      ILOC6=ILOC+6
      WRITE(IOOUT,10) (REGION(J),J=ILOC,ILOC6),SEASON(2*NSEASN-1),
     1SEASON(2*NSEASN),HOUR(2*NTIME-1),HOUR(2*NTIME)
C
C          READ THE THERMODYNAMIC DATA FOR REGION LOCAT AT NTIME
C                DURING NSEASN.
C
    3 DO 4 J=1,22
      READ(NCLIMT,11) NCLASS,(DATA(K),K=1,18)
C
C          CONVERT FROM METERS TO KILOMETERS
C
      DATA(6)=0.001*DATA(6)
      DATA(10)=0.001*DATA(10)
C
C          IF NPRT GT 0, PRINT THE THERMODYNAMIC DATA
C
      IF(NPRT.LE.0) GO TO 4
      WRITE(IOOUT,12) NCLASS,(DATA(K),K=1,18)
    4 CONTINUE
C
C          EXTRACT THE VALUES OF TEMP, PRESS, RH, AH, DP, VIS, AND
C                WNDVEL.
C
```

```
      TEMP=DATA(2)
      DP=DATA(3)
      AH=DATA(4)
      RH=DATA(5)
      VIS=DATA(6)
      PRESS=DATA(7)
      WNDVEL=DATA(8)
C
C          DETERMINE THE VALUE OF IPASCT
C
      IPASCT=1
      FREQ=DATA(13)
      DO 5 J=2,6
      IF(DATA(J+12).LE.FREQ) GO TO 5
      FREQ=DATA(J+12)
      IPASCT=J
    5 CONTINUE
C
C          GET TO THE WIND DATA ON NCLIMT
C
      NSKIP=(NSKIP-LSKIP)/44
      NSKIP=44*(15-NSKIP)+22*NSKIP
      DO 16 J=1,NSKIP
      READ(NCLIMT,9) A
   16 CONTINUE
C
C          IF NPRT GT 0, PRINT A HEADING FOR THE WIND DATA
C
      IF(NPRT.LE.0) GO TO 6
      WRITE(IOOUT,13) (DIR(J),J=1,13)
C
C          READ THE WIND DATA FOR REGION LOCAT AT NTIME DURING NSEASN
C
    6 DO 7 J=1,22
      READ(NCLIMT,14) NCLASS,(DATA(K),K=1,14)
C
C          IF NPRT GT 0, PRINT THE WIND DATA
C
      IF(NPRT.LE.0) GO TO 7
      WRITE(IOOUT,17) NCLASS,(DATA(K),K=1,14)
    7 CONTINUE
C
C          DETERMINE THE VALUE OF WINDIR
C
      NDIR=1
      FREQ=DATA(2)
      DO 8 J=2,12
      IF(DATA(J+1).LE.FREQ) GO TO 8
      FREQ=DATA(J+1)
      NDIR=J
    8 CONTINUE
      WINDIR=30*NDIR-15
C
C          RETURN FROM CLIMAT
C
      RETURN
C
C          FORMAT STATEMENTS
C
    9 FORMAT(A1)
   10 FORMAT(25H1 EOSAEL CLIMATOLOGY FOR ,7A4,8H DURING ,2A4,4H AT ,2A4,
     17H(LST). //126H CLASS  FREQCY   MEAN   MEAN   MEAN   MEAN   MEAN
     2 MEAN MEAN/STDEV MEAN  MEAN/STDEV FREQCY FREQCY FREQCY FREQCY FREQ
     3CY FREQCY/126H  NO.    CLASS    TEMP    DP     AH     RH     VIS  P
     4RESS   WNDVEL   CLDHT   CLDCVR      A      B      C      D      E
     5     F  /126H            (%)      (C)     (C) (GM/CU.M) (%)   (KM)   (
     6MB)   (MPS)    (KM)      (%)        (%)     (%)     (%)     (%)     (%)
     7   (%) /)
   11 FORMAT(6X,I3,5F5.1,F7.0,F6.1,2F5.1,F6.0/8F5.1)
   12 FORMAT(I5,F9.1,4F7.1,F7.3,F7.1,F4.1,1H/,F4.1,F7.3,F6.1,1H/,F5.1,6F
```

```
     17.1)                                                           CLI02110
13 FORMAT(1H0/7H CLASS ,14(7H FREQCY)/14H  NO.    CLASS,13(7H WNDDIR)CLI02120
     1/10X,3H(%),1X,13(2X,A4,1X)/14X,13(3X,3H(%),1X)/)                 CLI02130
14 FORMAT(6X,I3,14F5.1)                                               CLI02140
15 FORMAT(18H1CLIMATOLOGY MODEL///39H  DEFINITIONS OF METEOROLOGICAL CLI02150
     1CLASSES//48H CLASS  1 = FOG, HAZE AND MIST WITH VIS LT 1 KM./53H CCLI02160
     2LASS  2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM./53H CLASS  3 =CLI02170
     3 FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM./48H CLASS  4 = FOG, HAZCLI02180
     4E AND MIST WITH VIS GE 7 KM./34H CLASS  5 = DUST WITH VIS LT 3 KM.CLI02190
     5/34H CLASS  6 = DUST WITH VIS GE 3 KM./53H CLASS  7 = DRIZZLE, RAICLI02200
     6N AND TSTMS WITH VIS LT 1 KM./58H CLASS  8 = DRIZZLE, RAIN AND TSTCLI02210
     7MS WITH 1 LE VIS LT 3 KM./58H CLASS  9 = DRIZZLE, RAIN AND TSTMS WCLI02220
     8ITH 3 LE VIS LT 7 KM./53H CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH CLI02230
     9VIS GE 7 KM./34H CLASS 11 = SNOW WITH VIS LT 1 KM./39H CLASS 12 = CLI02240
     ASNOW WITH 1 LE VIS LT 3 KM./39H CLASS 13 = SNOW WITH 3 LE VIS LT 7CLI02250
     B KM./34H CLASS 14 = SNOW WITH VIS GE 7 KM./59H CLASS 15 = NO WEATHCLI02260
     CER AND ABSOLUTE HUMIDITY LT 10 GM/CU M./59H CLASS 16 = NO WEATHER CLI02270
     DAND ABSOLUTE HUMIDITY GE 10 GM/CU M./52H CLASS 17 = VIS LT 1 KM ANCLI02280
     ED CEILING HEIGHT LT 300 M./53H CLASS 18 = VIS LT 3 KM AND CEILING CLI02290
     FHEIGHT LT 1000 M./36H CLASS 19 = CEILING HEIGHT LT 300 M./37H CLASCLI02300
     GS 20 = CEILING HEIGHT LT 1000 M./23H CLASS 21 = NO CEILING./36H CLCLI02310
     HASS 22 = ALL CONDITIONS COMBINED.)                               CLI02320
17   FORMAT(I5,F9.1,13F7.1)                                           CLI02330
C                                                                      CLI02340
      END                                                              CLI02350
```

```
      PROGRAM AGAUS
C                            PROGRAM AGAUS                                  AGX00020
C                                                                           AGX00030
C          REVISION DATE 22 JANUARY, 1982                                   AGX00040
C                                                                           AGX00050
C                                                                           AGX00060
C     PURPOSE:                                                              AGX00070
C             TO CALCULATE EXTINCTION COEFFICIENTS, ETC., AND PRODUCE       AGX00080
C     LEGENDRE EXPANSION COEFFICIENTS, PHASE FUNCTIONS AND (OPTIONALLY)     AGX00090
C     SCATTERING FRACTIONS UNDER A VARIETY OF CONDITIONS AND AEROSOL        AGX00100
C     DISTRIBUTIONS AT ONE OR MORE WAVELENGTHS.  THE PHASE FUNCTION         AGX00110
C     IS NORMALIZED TO 4 PI OMEGA ZERO AND MAY BE RENORMALIZED BY           AGX00120
C     DIVISION BY THE APPROPRIATE CONSTANT(S).                              AGX00130
C                                                                           AGX00140
C     **********************  INPUT  *************************              AGX00150
C                                                                           AGX00160
C     CARD 1 - IDENTIFIER - 80 ALPHA CHARACTERS                             AGX00170
C                                                                           AGX00180
C     CARD 2 - INTEGER CONTROL PARAMETERS: NWAVE, NINDX, IW, IDSTP,         AGX00190
C              NRADI, IT, MORTE, IANG, IEO, NEOU                            AGX00200
C              FORMAT (10I5)                                                AGX00210
C   NWAVE: IS THE NO. OF WAVELENGTHS, OR REL. HUMIDITY VALUES TO BE         AGX00220
C     TREATED IN THIS RUN. SEE COMMENTS CIRCA READ OF WAVE, DWAVE, ETC.     AGX00230
C     N.B. NWAVE MUST BE .LE. 10 - TO CHANGE THE NUMBER OF WAVELENGTHS      AGX00240
C     CHANGE THE FIRST INDEX OF ARRAY OUT(I,J) TO AGREE WITH NWAVE.         AGX00250
C   NINDX: IS THE NBR OF AEROSOL COMPONENTS WHICH WILL HAVE DIFFERENT       AGX00260
C     OPTICAL CONSTANTS, MASS DENSITIES OR MASS CONCENTRATIONS.             AGX00270
C   IW: =0 WILL SET THE REFRACTIVE INDEX OF THE AEROSOL EQUAL               AGX00280
C     TO THAT OF WATER AT THE INPUT WAVELENGTH AND TEMP.  IF IW .NE. 0      AGX00290
C     AND HANEL'S GROWTH FACTOR IS ZERO (EMUA=0. - CARD 5), THEN THE        AGX00300
C     INPUT REFRACTIVE INDEX (EMA,CAYA) WILL BE USED FOR THE AEROSOL.       AGX00310
C     OTHERWISE THE REFRACTIVE INDEX IS ADJUSTED PER HANEL (SEE BELOW).     AGX00320
C   IDSTP: IDENTIFIES TYPE OF AEROSOL SIZE DISTRIBUTION TO BE USED.         AGX00330
C   NRADI: NO. OF PARTICLE RADII TO BE EXPECTED FOR IDSTP=0 OR 3;           AGX00340
C     THE INPUT VALUE OF NRADI IS IGNORED FOR IDSTP NOT ZERO OR 3.          AGX00350
C     NRADI MUST BE .LE. 1+2**JDIMCK(2) - C.F. BLOCK DATA                   AGX00360
C   IT: IS THE NUMBER OF GAUSS-LEGENDRE ANGLES (ORDER OF EXPANSION)         AGX00370
C     IF ONLY EXTINCION COEFFICIENTS, ETC. ARE DESIRED, I.E. NOT PHASE      AGX00380
C     FUNCTIONS, THEN SET -IT- EQUAL TO ONE.                                AGX00390
C   MORTE:=12345 WILL CAUSE PRINTS OF MIE EFFICIENCY FACTORS AT             AGX00400
C     EVERY VALUE OF PARTICLE RADIUS USED IN THE MIE CALCULATIONS;          AGX00410
C     SET MORTE = 0 IF SUCH PRINTS ARE NOT DESIRED.                         AGX00420
C  IANG:=0 FOR COMPUTATIONS OF PHASE FN. AT -IT- GAUSS LEGENDRE             AGX00430
C     QUADRATURE ANGLES; IANG=1 FOR COMPUTATIONS OF PHASE FN AT             AGX00440
C     -IT- EQUALLY SPACED ANGLES BETWEEN 0 AND 180 DEGREES.                 AGX00450
C     IANG=2 WILL ALLOW -IT- USER SUPPLIED ANGLES TO BE READ -              AGX00460
C     FORMAT (16F5.1).  THIS REQUIRES AT LEAST ONE CARD OF TYPE 2A.         AGX00470
C     IF IANG.GT.0 NO LEGENDRE COEFFICIENTS WILL BE GENERATED.              AGX00480
C  IEO=1,2,3,4 WILL CONSTRUCT A PHASE FUNCTION FILE (ON NEOU).              AGX00490
C     IEO=1   65 PREDETERMINED ANGLES    INDIVIDUAL WAVELENGTHS ONLY        AGX00500
C     IEO=2   65 PREDETERMINED ANGLES    COMPOSITE WAVELENGTH ONLY          AGX00510
C     IEO=3   USER INPUT ANGLES          INDIVIDUAL WAVELENGTHS ONLY        AGX00520
C     IEO=4   USER INPUT ANGLES          COMPOSITE WAVELENGTH ONLY          AGX00530
C     IEO=5   65 PREDETERMINED ANGLES    INDIVIDUAL & COMPOSITE WAVELENGTHAGX00540
C                                        THE COMPOSITE WILL BE THE LAST DAAGX00550
C                                        SET WRITTEN ON UNIT -NEOU- .       AGX00560
C     FOR USER INPUT ANGLES SEE -IT- AND IANG ABOVE.  THE COMPOSITE         AGX00570
C     VALUES ARE SIMPLE AVERAGES OVER THE NUMBER OF WAVELENGTHS.            AGX00580
C     THIS FILE WILL CONTAIN THE FOLLOWING INFORMATION:                     AGX00590
C     1) ANGLES (65 MAX) - FORMAT(11(F6.2,1X))                              AGX00600
C     2)NBR OF ANGLES, PHASE FUNCTION IDENTIFIER (=0= IMPLIED USER INPUTAGX00610
C     IN EOSAEL), WAVELENGTH (UM), SINGLE SCATTERING ALBEDO, EXTINCTION AGX00620
C     AND SCATTERING COEFFICIENTS IN INVERSE KM -                           AGX00630
C     FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))                         AGX00640
C     3) PHASE FUNCTION AT ANGLES SPECIFIED ABOVE.  N.B. THE PHASE FUNCTAGX00650
C     AS WRITTEN OUT HERE IS NORMALIZED TO 4 PI OMEGA ZERO;  THE ROUTINEAGX00660
C     IN EOSAEL WILL RENORMALIZE THE PHASE FUNCTION TO ONE.                 AGX00670
C     FORMAT (6(E12.6,1X))                                                  AGX00680
C  NEOU= UNIT NUMBER UPON WHICH EOSAEL PHASE FUNCTION IS TO BE STORED.      AGX00690
C                                                                           AGX00700
```

```
C       CARD 2A - USER SUPPLIED SET OF -IT- ANGLES  FORMAT (16F5.1)          AGX00710
C               16 VALUES PER CARD, MORE THAN 1 CARD MAY BE NEEDED.          AGX00720
C                 THIS CARD IS ONLY NEEDED WHEN IANG=2.                      AGX00730
C                                                                            AGX00740
C       CARD 3 - DISTRIBUTION PARAMETERS: ONLY ONE TYPE PER RUN.             AGX00750
C                 FORMAT (6E12.6)  - READ IN AGXP1                           AGX00760
C       *****  ALL DIMENSIONS ARE IN MICRONS  *****                          AGX00770
C         TYPE 0. USER SUPPLIED - NRADI CARDS, ONE VALUE OF RADIUS AND
C                 NUMBER DENSITY PER CARD.  IT IS SUGGESTED THAT DELTA BE
C                 INPUT NO GREATER THAN .001 IN ORDER TO FORCE THE MAXIMUM
C                 NUMBER OF RADII TO BE USED DUE TO THE POSSIBLE IRREGULAR
C                 NATURE OF THIS DISTRIBUTION.
C                 R(I), FF(I), I = 1, NRADI
C         TYPE 1. LOG-NORMAL                                                 AGX00810
C                 RBAR, SIGMA, RLO, RHI                                      AGX00820
C         TYPE 2. DOUBLE EXPONENTIAL                                         AGX00830
C                 RLO, RHI, CUE, A, B                                        AGX00840
C         TYPE 3. DEIRMENDJIAN MODEL C                                       AGX00850
C                 - NO INPUT -                                               AGX00860
C         TYPE 4. POWER LAW (JUNGE)                                          AGX00870
C                 RLO, RHI, CUE, A                                           AGX00880
C         TYPE 5. MODIFIED GAMMA                                             AGX00890
C                 RLO, RHI, RC, ALF, GAM                                     AGX00900
C         TYPE 6. MODIFIED GAMMA FOG MODEL                                   AGX00910
C                 RLO, RHI, RC, ALF, GAM, ELWC                               AGX00920
C         TYPE 7. POWER LAW                                                  AGX00930
C                 VIS                                                        AGX00940
C         TYPE 8. CONTINENTAL BIMODAL                                        AGX00950
C                 - NO INPUT -                                               AGX00960
C         TYPE 9. MARITIME BIMODAL                                           AGX00970
C                 - NO INPUT -                                               AGX00980
C         TYPE 10.URBAN BIMODAL                                              AGX00990
C                 - NO INPUT -                                               AGX01000
C         TYPE 11.USER SUPPLIED BIMODAL                                      AGX01010
C                 FOA, RBARA, SGA, FOC, RBARC, SGC                           AGX01020
C         TYPE 12.MARSHALL-PALMER RAIN MODEL                                 AGX01030
C                 RAIN                                                       AGX01040
C                                                                            AGX01050
C       CARD 4 - CONTROL PARAMETERS: FORMAT (6E12.6)                         AGX01060
C                 WAVE, DWAVE, RELHUM, DENSH, TEMP, DELTA                    AGX01070
C                 FOR LOOPING OVER RELATIVE HUMIDITY ADD                     AGX01080
C                 NWAVE-1 CARDS CONTAINING RELHUM,TEMP - FORMAT(2E12.6)      AGX01090
C                 SEE DWAVE BELOW.                                           AGX01100
C     WAVE: IS WAVELENGTH IN MICROMETERS.                                    AGX01110
C     DWAVE: IS THE WAVELENGTH INCREMENT IN MICROMETERS. IF DWAVE IS         AGX01120
C       LESS THAN 1.E-4, A SPECIAL CASE APPLIES USED FOR LOOPING OVER        AGX01130
C       NWAVE VALUES OF RELHUM: THE FIRST TIME THIS CARD IS READ IT          AGX01140
C       MUST CONTAIN WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA; THE SECOND AND      AGX01150
C       SUBSEQUENT TIMES IT MUST ONLY HAVE RELHUM,TEMP ON IT.  THIS          AGX01160
C       ALSO REQUIRES REPETITION OF CARD 5.                                  AGX01170
C     RELHUM: IS RELATIVE HUMIDITY IN PERCENT.                               AGX01180
C     DENSH: IS PARTICLE NUMBER PER CUBIC CENTIMETER.                        AGX01190
C       USER-SUPPLIED VALUE OF DENSH WILL BE IGNORED FOR IDSTP=3 OR GT 6     AGX01200
C       BECAUSE THOSE DISTRIBUTIONS CARRY PRE-DETERMINED DENSITY VALUES.     AGX01210
C       ALSO, IF DENSH IS LESS THAN 1E-4, THE PARTICLE NUMBER DENSITY        AGX01220
C       WILL BE CALCULATED FROM MASS DENSITY AND MASS CONCENTRATION.         AGX01230
C     TEMP: IS THE TEMPERATURE OF THE ATMOSPHERE IN DEGREES C.               AGX01240
C     DELTA:  IS THE CONVERGENCE CRITERION WITHIN A PARTICULAR SIZE          AGX01250
C       RANGE INTERVAL: HALVING IS TERMINATED WHEN THE QUANTITY DEL          AGX01260
C       IS LESS THAN DELTA. **N.B.** THE AMOUNT OF CPU TIME USED BY          AGX01270
C       THIS PROGRAM IS CLOSELY CONNECTED WITH DELTA.  THE SMALLER DELTA     AGX01280
C       IS THE LARGER YOUR RUN TIME WILL BE.  IT IS SUGGESTED THAT           AGX01290
C       DELTA BE SET EQUAL TO .001 FOR MOST RUNS.                            AGX01300
C                                                                            AGX01310
C       CARD 5 -  OPTICAL AND PHYSICAL DATA: FORMAT (4F10.5,E15.7)           AGX01320
C                 EMA, CAYA, EMUA, RHOA, CONC                                AGX01330
C       REPEAT NWAVE*NINDX TIMES: IF IDSTP=6 THIS CARD IS NOT NEEDED.        AGX01340
C     EMA: IS THE REAL PART OF THE INDEX OF REFRACTION OF DRY AEROSOL.       AGX01350
C     CAYA: IS THE IMAGINARY PART OF REFRACTIVE INDEX FOR DRY AEROSOL.       AGX01360
C       ***** CAYA IS ASSUMED TO BE NEGATIVE  ****                           AGX01370
```

```
C      **** DO NOT ENTER CAYA WITH A NEGATIVE SIGN !!!!! *****
C    EMUA:  IS HANEL'S GROWTH FACTOR (MU-BAR)/ACCRETION COEF.
C    RHOA:  IS THE MASS DENSITY(SP. GRAV) OF DRY AEROSOL.
C    CONC:  IS THE MASS CONCENTRATION(GM/CC) OF DRY AEROSOL.
C
C      **********************  END INPUT  *********************
C
C      ****************** MISCELLANEOUS INFO ******************
C
C      THE INPUT AND OUTPUT UNITS, ALONG WITH A EXTRA, CURRENTLY UNUSED,
C      UNIT (NUNIT) ARE ASSIGNED VALUES IN THE BLOCK DATA SUBROUTINE.
C
C      REL. HUMIDITY TREATMENT PER G. HANEL/1976 ADV. IN GEOPHYS.
C
C      FOR DIMENSION SIZES REFER TO THE BLOCK DATA SUBROUTINE.  THERE
C      ALSO IS A ERROR ROUTINE (DIMER) THAT CKS ON YOUR DIMENSIONS.
C
C      SCATTERING FRACTIONS REQUIRE THAT NUNIT BE ASSIGNED AND A
C      SIMPLE CHANGE BE MADE IN SUBROUTINE AGXP3:  FOR FURTHER INFO
C      REFER TO THAT SUBROUTINE.
C
C      THE FUNCTION ATAN2(SQRT(1.-C(I)**2),C(I)) IS EQUIVALENT
C      TO ARCOS(C(I)).
C
C      **************** ACKNOWLEDGEMENTS **********************
C
C        THIS PROGRAM HAS BEEN CONSTRUCTED BY THE ATMOSPHERIC SCIENCES
C      LABORATORY AND NEW MEXICO STATE UNIVERSITY, DEPT OF PHYSICS.  THE
C      FOLLOWING PEOPLE HAVE PARTICIPATED IN THIS UNDERTAKING:
C
C                DR. A.U. MILLER          NMSU
C                DR. R.C. SHIRKEY         ASL
C                DR. G.H. GOEDECKE        NMSU
C                MR. E.J. BURLBAW         NMSU
C
C      THE POINT OF CONTACT IS R.C. SHIRKEY, ASL.  PHONE (505) 678-5470
C      OR AV 258-5470.
C
C      *************************************************************
      REAL KEXTT,KSCAT,KBAKT
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
     +,NRADI,PI,IDSTP,NKG,NHALV,NI
      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
      DIMENSION OUT(10,4),NTITLE(40)
C     *****  READ AND WRITE IDENTIFIER  *****
      READ (IOIN,88) (NTITLE(I),I=1,40)
 88   FORMAT (40A2)
      WRITE (IOUT,89) (NTITLE(I),I=1,40)
 89   FORMAT(1H1,40A2//)
C     ***** READ INTEGER CONTROL PARAMETERS FOR THIS RUN  *****
      READ (IOIN,103) NWAVE,NINDX,IW,IDSTP,NRADI,IT,MQRTE,IANG,IEO,NEOU
C     ERROR CHECKS
      IF (IT.LE.0) IT=1
      IF (IT.GT.JDIMCK(1)) CALL DIMER(1)
      IF(JDIMCK(1).LT.65) WRITE(IOUT,129)
C     CHECK FOR CONFLICTING EOSAEL OPTIONS
      IF ((IEO.EQ.2.OR.IEO.EQ.5).AND.NWAVE.EQ.1) IEO=1
      IF ((IEO.EQ.4.OR.IEO.EQ.5).AND.NWAVE.EQ.1) IEO=3
      IF (IEO.GT.1.AND.IT.GT.65) GO TO 20
C     EOSAEL OPTION
      IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) IT=65
      JDIMCK(3)=1+2**JDIMCK(2)
      IF (IDSTP.GT.12) GO TO 1
      IF (NWAVE.EQ.0) NWAVE=1
      IF (IDSTP.EQ.12) IW=1
      IF (NINDX.LT.1.OR.IDSTP.EQ.6.OR.IDSTP.EQ.12) NINDX=1
      WRITE (IOUT,104) NWAVE,NINDX,IW,IDSTP,NRADI,IT,MQRTE,IANG,IEO,NEOU
      IF (IW.EQ.0) WRITE (IOUT,122)
```

```
C     INITIALIZE QUANTITIES USED IN SUMMATIONS                          AGX02080
      DO 2 I=1,IT                                                       AGX02090
      OLT(I)=0.                                                         AGX02100
    2 PSUMT(I)=0.E0                                                     AGX02110
      WAVAVG=0.                                                         AGX02120
      ALBDOT=0.                                                         AGX02130
      KEXTT=0.E0                                                        AGX02140
      KSCAT=0.E0                                                        AGX02150
      KBAKT=0.E0                                                        AGX02160
      CATTN=0.E0                                                        AGX02170
      ITT=IT-1                                                          AGX02180
      PI=3.14159265358898E+00                                           AGX02190
      IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) GO TO 3                     AGX02200
      IF ((IANG.EQ.1).OR.(IANG.EQ.2)) GO TO 3                           AGX02210
C     WHEN IANG=0 ROUTINE GUSET IS CALLED TO SET-UP THE ABSCISSAE AND   AGX02220
C     WEIGHTS USED FOR CALCULATING THE PHASE-FUNCTION AT -IT- POINTS    AGX02230
C     USED FOR NUMERICAL INTEGRATION VIA GAUSS-LEGENDRE QUADRATURE AND  AGX02240
C     THE PHASE FUNCTION EXPANSION COEFS, OL( ).                        AGX02250
C     THE WEIGHTS ARE PLACED IN THE ARRAY W( ), AND THE COSINES OF THE  AGX02260
C     ANGLES ARE PLACED IN THE ARRAY C( ).                              AGX02270
      CALL GUSET(IT)                                                    AGX02280
      IF (ITT.LT.3) ITT=3                                               AGX02290
      GO TO 7                                                           AGX02300
3     CALL ANGLE (PI,IANG,IT)                                           AGX02310
C     SUBROUTINE ANGLE IS CALLED WHEN IANG=1 OR 2 TO SET UP THE         AGX02320
C     ANGLES AT WHICH PHASE FUNCTIONS WILL BE CALCULATED.  ANGLES       AGX02330
C     GO INTO ARRAY W( ) AND COSINES IN C( ).                           AGX02340
7     CONTINUE                                                          AGX02350
C     WRITE ANGLES FOR EOSAEL DATA FILE                                 AGX02360
      IF (IEO.LE.0) GO TO 21                                            AGX02370
      DO 22 I=1,IT                                                      AGX02380
22    C(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI                         AGX02390
      ITP1=IT+1                                                         AGX02400
      IF (ITP1.GT.JDIMCK(1)) ITP1=IT                                    AGX02410
      IF (IT.LT.65.AND.JDIMCK(1).GT.65) C(ITP1)=999.99                  AGX02420
      WRITE (NEOU,125)  (C(I),I=1,ITP1)                                 AGX02430
      DO 23 I=1,IT                                                      AGX02440
23    C(I)=COS(C(I)*PI/180.)                                            AGX02450
21    CONTINUE                                                          AGX02460
C     DETERMINE DETAILS OF AEROSOL SIZE-DISTRIBUTION VIA AGXP1          AGX02470
      CALL AGXP1(DENS,FSUM,VOL,JDIMCK)                                  AGX02480
      IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) ELWC=DENS                          AGX02490
C     DRYVOL IS THE AVERAGE VOLUME OF THE DRY AEROSOL PARTICLES IN      AGX02500
C     CUBIC MICROMETERS.                                                AGX02510
      DRYVOL=VOL                                                        AGX02520
C     *** READ INPUT PARAMETERS ***                                     AGX02530
      READ (IOIN,105) WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA                AGX02540
      IF (NWAVE.EQ.1) DWAVE=0.E0                                        AGX02550
      WRITE (IOUT,106) WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA               AGX02560
      IF (NINDX.GT.1) WRITE (IOUT,107) NINDX                            AGX02570
      IF ((DWAVE.LT.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,108) NWAVE     AGX02580
      IF ((DWAVE.GE.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,109) NWAVE     AGX02590
      IF (DENSH.LT.1E-04) WRITE (IOUT,110)                              AGX02600
      ENWAV= FLOAT(NWAVE)                                               AGX02610
      IF (DWAVE.LT.1.E-4) GO TO 8                                       AGX02620
      WAVE=WAVE-DWAVE                                                   AGX02630
    8 DO 9 NWV=1,NWAVE                                                  AGX02640
      IF (DWAVE.GT.1.E-4) GO TO 10                                      AGX02650
      IF (NWV.EQ.1) GO TO 11                                            AGX02660
      READ (IOIN,105) RELHUM,TEMP                                       AGX02670
      GO TO 11                                                          AGX02680
   10 WAVE=WAVE+DWAVE                                                   AGX02690
   11 VOL=DRYVOL                                                        AGX02700
C     DETERMINE WHETHER THE USER SUPPLIED PARTICLE NUMBER DENSITY DENSH AGX02710
C     SHOULD BE OVERIDDEN BECAUSE THE CHOSEN IDSTP CASE HAS FIXED       AGX02720
C     PARAMETERS, AND/OR IF NUMBER DENSITIES ARE TO BE CALCULATED LATER AGX02730
C     FROM THE AVG PARTICLE VOLUME, MASS DENSITY, AND MASS CONCENTRATIONAGX02740
      LLLL=0                                                            AGX02750
      IF (IDSTP.EQ.6) GO TO 12                                          AGX02760
      IF (IDSTP.EQ.3.OR.IDSTP.GE.7) LLLL=1                              AGX02770
```

```
      IF (LLLL.EQ.1) GO TO 12                                           AGX02780
      IF (DENSH.LE.1.E-4) GO TO 12                                      AGX02790
      LLLL=1                                                            AGX02800
      DENS=DENSH                                                        AGX02810
   12 CONTINUE                                                          AGX02820
C     RESTRICT RELATIVE HUMIDITY TO MAX OF 99 PERCENT.                  AGX02830
      IF (RELHUM.GE.99.E+00) RELHUM=99.0E+00                            AGX02840
      WRITE (IOUT,111) RELHUM,WAVE                                      AGX02850
      IF (DENS.EQ.0.0) DENS=1.0E+00                                     AGX02860
      GNU=1.0E+04/WAVE                                                  AGX02870
      IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) DENS=ELWC                          AGX02880
C     DENS IS USED AS AN ALIAS TO PASS ELWC TO ROUTINE AGXP2.           AGX02890
      CALL AGXP2(RELHUM,CTSUM,CSSUM,CRSUM,VOL,TMASS,DENS,QATTN,TEMP,    AGX02900
     1DELTA,NINDX,IW,OLSTAR,OM2,LLLL,IT,WAVE,EM,CAY,EMM,MQRTE,PFNZRO)   AGX02910
      LMAX=3*IFIX(2.E+0*PI*EMM*R(NRADI)/WAVE)                           AGX02920
      IF (LMAX.GT.IT) WRITE (IOUT,112) LMAX,IT                          AGX02930
   13 CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,WAVE,EM,CAY,EMM,IT,0, AGX02940
     +IANG)                                                             AGX02950
C     SUM QUANTITIES OVER INDEX NWV.                                    AGX02960
      DO 14 IK=1,IT                                                     AGX02970
      OLT(IK)=OLT(IK)+OL(IK)                                            AGX02980
   14 PSUMT(IK)=PSUMT(IK)+PSUM(IK)                                      AGX02990
C     ALBDOT BECOMES THE TOTAL SINGLE SCATTERING ALBEDO                 AGX03000
C     KEXTT BECOMES THE TOTAL EXTINCTION COEF. (PER KILOMETER)          AGX03010
C     KSCAT BECOMES THE TOTAL SCATTERING COEF. (PER KM)                 AGX03020
C     KBAKT BECOMES THE TOTAL BACK-SCATTERING(RADAR) COEF (PER KM)      AGX03030
C     ARRAY OUT(,) HOLDS SOME QUANTITIES FOR LATER PRINTOUTS            AGX03040
      ALBDO=CSSUM/CTSUM                                                 AGX03050
      ALBDOT=ALBDOT+ALBDO                                               AGX03060
      KEXTT=KEXTT+CTSUM                                                 AGX03070
      KSCAT=KSCAT+CSSUM                                                 AGX03080
      KBAKT=KBAKT+CRSUM                                                 AGX03090
      CATTN=CATTN+QATTN                                                 AGX03100
      WAVAVG=WAVAVG+WAVE                                                AGX03110
      OUT(NWV,1)=WAVE                                                   AGX03120
      OUT(NWV,2)=RELHUM                                                 AGX03130
      OUT(NWV,3)=TMASS*1.E5                                             AGX03140
      OUT(NWV,4)=CTSUM                                                  AGX03150
      IF ((NWAVE.GT.1).AND.(DWAVE.GE.1.E-04)) WRITE (IOUT,113) NWV      AGX03160
      IF ((NWAVE.GT.1).AND.(DWAVE.LT.1.E-04)) WRITE (IOUT,114) NWV      AGX03170
C     EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH, SINGLE SCATTERING AGX03180
C     ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR         AGX03190
C     INDIVIDUAL WAVELENGTHS.                                           AGX03200
      IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,127)            AGX03210
     +                          IT,WAVE,ALBDO,CTSUM,CSSUM               AGX03220
C     EOSAEL OPTION: WRITE PHASE FUNCTION FOR INDIVIDUAL WAVELENGTHS.   AGX03230
      IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,128)            AGX03240
     +   (PSUM(I),I=1,IT)                                               AGX03250
9     CONTINUE                                                          AGX03260
C     END OF NWAVE LOOP                                                 AGX03270
      IF (NWAVE.LE.1) GO TO 19                                          AGX03280
C     DIVIDE BY NBR OF VALUES OF NWV TO GET AVERAGED RESULTS            AGX03290
      DO 16 I=1,IT                                                      AGX03300
      OL(I)=OLT(I)/ENWAV                                                AGX03310
      PSUM(I)=PSUMT(I)/ENWAV                                            AGX03320
16    CONTINUE                                                          AGX03330
      ALBDOT=ALBDOT/ENWAV                                               AGX03340
      KEXTT=KEXTT/ENWAV                                                 AGX03350
      KSCAT=KSCAT/ENWAV                                                 AGX03360
      KBAKT=KBAKT/ENWAV                                                 AGX03370
      CATTN=CATTN/ENWAV                                                 AGX03380
      WAVAVG=WAVAVG/ENWAV                                               AGX03390
      WRITE (IOUT,117) NWAVE                                            AGX03400
      WRITE (IOUT,118)                                                  AGX03410
      DO 18 J=1,NWAVE                                                   AGX03420
18    WRITE (IOUT,119) (OUT(J,JJ),JJ=1,4)                               AGX03430
      WRITE (IOUT,123) KEXTT,KSCAT,KBAKT,CATTN,ALBDOT                   AGX03440
      CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,WAVE,EM,CAY,EMM,IT,1, AGX03450
     +IANG)                                                             AGX03460
C     EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH ,SINGLE SCATTERING AGX03470
```

```
C     ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR
C     COMPOSITE VALUES.
      IF (IEO.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,127)
     +                          IT,WAVAVG,ALBDOT,KEXTT,KSCAT
C     EOSAEL OPTION: WRITE COMPOSITE PHASE FUNCTION.
      IF (IEO.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,128)
     +    (PSUM(I),I=1,IT)
      GO TO 19
    1 WRITE (IOUT,120) IDSTP
   19 WRITE (IOUT,126)
      STOP
20    WRITE (IOUT,124)
      STOP
  103 FORMAT (10I5)
  104 FORMAT(1H ,78HINTEGER CONTROL PARAMETERS: NWAVE NINDX IW IDSTP NRA
     +DI IT MQRTE  IANG IEO NEOU/,1X,29X,2(I2,4X),I1,3X,I2,
     +3X,I3,1X,I3,1X,I5,1X,I5,2(2X,I2))
  105 FORMAT (6E12.6)
  106 FORMAT(1H ,/,17H INPUT PARAMETERS/,1X,6X,9HWAVE =    ,E12.6,8H MICR
     +ONS/,1X,6X,9HDWAVE =  ,E12.6,8H MICRONS/,1X,6X,9HRELHUM = ,E12.6,
     +8H PERCENT/,1X,6X,9HDENSH =  ,E12.6,13H PARTICLES/CC/,1X,6X,
     +9HTEMP =    ,E12.6,6H DEG C/,1X,6X,19HDELTA (CONVERGENCE ,
     +13HCRITERION) = ,E12.6)
  107 FORMAT (/,1H ,29HLOOPING OPTION IN EFFECT FOR ,I2,
     +19H AEROSOL COMPONENTS)
  108 FORMAT (/,1H ,39HRELATIVE HUMIDITY OPTION IN EFFECT FOR ,I2,
     +7H VALUES)
  109 FORMAT (/,1H ,40HWAVELENGTH LOOPING OPTION IN EFFECT FOR ,I2,
     +12H WAVELENGTHS)
  110 FORMAT(1H ,52H***  PARTICLE NUMBER DENSITY WILL BE CALCULATED FROM
     +,41H MASS DENSITY AND MASS CONCENTRATION  ***)
  111 FORMAT (1H1,//,1X,33HRELATIVE HUMIDITY FOR THIS RUN = ,F6.2,
     +25H  PERCENT.  WAVELENGTH = ,F10.3,8H MICRONS,/)
  112 FORMAT (/49H *** WARNING *** OPTIMAL PF EXPANSION ORDER OF   ,I3,
     +          22H  EXCEEDS INPUT IT = ,I3,24H.  PF VALUES SHOULD BE ,
     +          15HUSED CAUTIOUSLY/)
  113 FORMAT (1H //,1X,40(1H*),3X,31HEND OF WAVELENGTH CYCLE NUMBER ,I3,
     +3X,40(1H*))
  114 FORMAT (1H //,1X,40(1H*),3X,38HEND OF RELATIVE HUMIDITY CYCLE NUMB
     +ER ,I3,3X,40(1H*))
117   FORMAT(1H1,/,47H SUMMARY OF RESULTS FOR THIS RUN AVERAGED OVER ,
     +I2,30H WAVELENGTH(S) ARE AS FOLLOWS://)
118   FORMAT(1H ,4X,48HWAVELENGTH    REL.HUMIDITY    AEROSOL MASS    K(,
     +  11HEXTINCTION),/1X,29H  (MICROMETERS)     (PERCENT)
     +  ,6H   (GM,25H/(SQ.CM-KM)       (PER KM)/)
  119 FORMAT (2F15.6,1P2E16.5)
  120 FORMAT (//13H *** IDSTP = ,I5,35H IS ILLEGAL. EXECUTION TERMINATED
     +          2H* //)
  122 FORMAT (/,1X,23H*** WATER ONLY CASE ***/)
123   FORMAT (/,20H EXTINCTION COEF. = ,5X,1PE13.7,9H (PER KM),/,
     120H SCATTERING COEF. = ,8X,1PE13.7,9H (PER KM),/,
     225H BACK-SCATTERING COEF. = ,3X,1PE13.7,9H (PER KM),/,
     321H ATTENUATION COEF. = ,7X,1PE13.7,13H SQ-METERS/MG,/,
     428H SINGLE SCATTERING ALBEDO = ,1PE13.7,/)
124   FORMAT(1H ,58H*** MORE THAN 65 ANGLES FOR EOSAEL OPTION - PGM TERM
     +INATED)
125   FORMAT (11(F6.2,1X))
126   FORMAT(1H1)
127   FORMAT(I2,1X,2H00,1X,F5.2,1X,F8.6,1X,2(E12.6,1X))
128   FORMAT(6(E12.6,1X))
129   FORMAT(1H ,23H**** AGAUS WARNING ****,/,1X,
     +  37HTHE ARRAY W IS ASSIGNED 65 VALUES IN ,/,1X,
     +  43HBLOCK DATA WHICH IS LARGER THAN ARRAY SIZE ,/,1X,
     +  47HYOU MAY BE CLOBBERING INSTRUCTIONS AND/OR DATA./)
      END
```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

5 6

```
      SUBROUTINE AGXP1 (DENS,FSUM,VOL,JDIMCK)                           AGA00010
      COMMON /PT1/  F(513),R(513),DR(8),RR(9),FF(514)                   AGA00020
     +,NRADI,PI,IDSTP,NKG,NHALV,NI                                      AGA00030
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                              AGA00040
      DIMENSION JDIMCK(3)                                               AGA00050
      EXTERNAL GAMMA                                                    AGA00060
      WRITE (IOUT,2)                                                    AGA00070
C     CHOOSE AND SET UP PARTICLE SIZE DISTRIBUTION                      AGA00080
      IF (IDSTP.NE.0) GO TO (12,15,16,17,19,19,17,20,21,22,24,26),IDSTP AGA00090
C**   TYPE 0: ARBITRARY USER-SUPPLIED DISTRIBUTION. NRADI VALUES OF
C     R(J) AND FF(J) MUST BE GIVEN, ONE PER CARD, AND READ IN ORDER
C     FROM SMALLEST RADIUS, RLO TO THE LARGEST.
C     NRADI MUST BE LESS THAN OR EQUAL TO 1+2**JDIMCK(2)
      WRITE (IOUT,7)                                                    AGA00140
      IF (NRADI.GT.JDIMCK(3)) CALL DIMER(2)                             AGA00160
      DO 9 J=1,NRADI                                                    AGA00170
    9 READ (IOIN,1) R(J),FF(J)                                          AGA00190
      RLO=R(1)
      DO 10 J=1,NRADI,5                                                 AGA00200
      JK=J+4                                                            AGA00210
      IF(JK.GT.NRADI)JK=NRADI                                           AGA00220
   10 WRITE (IOUT,11)(R(K),FF(K),K=J,JK)                                AGA00230
      WRITE (IOUT,11)                                                   AGA00240
      FF(NRADI+1)=FF(NRADI)                                             AGA00250
      RR(1)=RLO                                                         AGA00260
      RR(2)=R(NRADI)                                                    AGA00270
      MIN=0                                                             AGA00280
      GO TO 28                                                          AGA00290
C**   TYPE 1: LOG-NORMAL DISTRIBUTION                                   AGA00300
   12 READ (IOIN,1) RBAR,SIGMA,RLO,RHI                                  AGA00310
C     SIGMA IS STANDARD DEVIATION, NOT LN(SIGMA)                        AGA00320
      SIGIN=SIGMA                                                       AGA00330
      SIGMA=ALOG(SIGMA)                                                 AGA00340
      A=ABS(1.E0/(2.5066283E0*SIGMA))                                   AGA00350
      IF ((RHI-RLO).LE.1.E-4) GO TO 13                                  AGA00360
      RR(1)=RLO                                                         AGA00370
      RR(3)=RHI                                                         AGA00380
      GO TO 14                                                          AGA00390
   13 RR(1)=RBAR*EXP(-4.E0*SIGMA)                                       AGA00400
      RR(3)=RBAR*EXP(4.E0*SIGMA)                                        AGA00410
   14 RR(2)=RBAR                                                        AGA00420
      MIN=1                                                             AGA00430
      WRITE (IOUT,3) RBAR,SIGIN,RLO,RHI                                 AGA00440
      AVOL=4.18879E0*(RBAR**3.E0)*EXP(4.5E0*SIGMA*SIGMA)                AGA00450
C     HERE AND ELSEWHERE, AVOL IS THE VOLUME OBTAINED VIA               AGA00460
C     ANALYTICAL INTEGRATION OVER THE LIMITS RLO =0 TO RHI =            AGA00470
C     INFINITY: THAT CAN ONLY BE DONE FOR A FEW IDSTP CASES.            AGA00480
      GO TO 28                                                          AGA00490
C** TYPE 2: DOUBLE EXPONENTIAL F(R)=CUE*A*EXP(-A*R)+(1-CUE)*B*EXP(-B*R).AGA00500
C       RESTRICTIONS: RHI.GT.RLO, B.GT.A.GE.0, 0.LE.CUE.LE.1.0.         AGA00510
   15 READ (IOIN,1) RLO,RHI,CUE,A,B                                     AGA00520
      WRITE (IOUT,4) RLO,RHI,CUE,A,B                                    AGA00530
      RR(1)=RLO                                                         AGA00540
      RR(3)=RHI                                                         AGA00550
      RR(2)=0.5E0*(RLO+RHI)                                             AGA00560
      MIN=1                                                             AGA00570
      GO TO 28                                                          AGA00580
C**   TYPE 3: DEIRMENDJIAN MODEL C. F(R) = 1.0, RLO.LE.R.LE.4*DELRD,    AGA00590
C     F(R)=(4*DELRD/R)**4, R.GE.(4*DELRD)                               AGA00600
C     NRADI IS READ IN EARLIER IN THE MAIN PROGRAM.                     AGA00610
   16 DENS=1.378E+04                                                    AGA00620
      DELRD=0.02E0                                                      AGA00630
      RLO=0.02E0                                                        AGA00640
      RHI=RLO+DELRD* FLOAT(NRADI-1)                                     AGA00650
      RR(1)=RLO                                                         AGA00660
      RR(3)=RHI                                                         AGA00670
      MIN=1                                                             AGA00680
      RR(2)=RLO+4.E0*DELRD                                              AGA00690
      GO TO 28                                                          AGA00700
C**   TYPE 4 AND TYPE 7: POWER LAW. F(R) = CUE*R**-A                    AGA00710
```

```
C     RLO.LE.R.LF.RHI; VIS=VISIBILITY IN KILOMETERS.
   17 IF (IDSTP.EQ.4) GO TO 18
C**   TYPE 7 PRESCRIBED PARAMETER.
      READ (IOIN,1) VIS
      RLO=0.1E0
      RHI=15.E0
      CUE=30.E0
      A=4.E0
      DENS=11.E0**(5.E0-ALOG10(VIS))
C**   TYPE 4 PRESCRIBED PARAMETERS.
   18 IF (IDSTP.EQ.4) READ (IOIN,1) RLO,RHI,CUE,A
      WRITE (IOUT,5) RLO,RHI,CUE,A,VIS
      RR(1)=RLO
      RR(3)=RHI
      RR(2)=(0.5E0*(RLO**(-A)+RHI**(-A)))**(-1.E0/A)
      MIN=1
      GO TO 28
C**   TYPE 5: MODIFIED GAMMA/GENERALIZED KHIRGIAN-MAZIN
C     F(R) = (R**ALF)*EXP(-ALF*((R/RC)**GAM)/GAM)
C     RLO.LE.R.LE.RHI.
C**   TYPE 6: SPECIAL CASE FOR WATER FOGS OR CLOUDS,
C     IN WHICH CASE ELWC IS LIQUID WATER CONTENT
C     IN GRAMS PER CUBIC CENTIMETER:
C     ELWC IS IGNORED IF IDSTP = 5.
   19 READ (IOIN,1) RLO,RHI,RC,ALF,GAM,ELWC
      IF (IDSTP.EQ.6) DENS=ELWC
      WRITE (IOUT,6) RLO,RHI,RC,ALF,GAM
      RR(1)=RLO
      RR(2)=RC
      RR(3)=RHI
      MIN=1
      B=ALF/(GAM*RC**GAM)
      AVOL=4.1888*B**(-3./GAM)*GAMMA((ALF+4.)/GAM)/GAMMA((ALF+1.)/GAM)
      GO TO 28
C**   TYPES 8,9,10: BIMODAL LOG-NORMAL DISTRIBUTIONS.
C     METHOD BELOW VALID FOR RBARC*EXP(-SGA).GT.RBARA*EXP(SGA)
C**   TYPE 8: CONTINENTAL BIMODAL.
   20 FOA=4.E03
      FOC=2.1E0
      SGA=0.74E0
      SGC=0.81E0
      RBARA=0.03E0
      RBARC=0.4E0
      GO TO 23
C**   TYPE 9: MARITIME BIMODAL.
   21 FOA=4.E02
      FOC=3.8E0
      SGA=0.68E0
      SGC=0.74E0
      RBARA=0.05E0
      RBARC=0.65E0
      GO TO 23
C**   TYPE 10: URBAN BIMODAL.
   22 FOA=2.E04
      FOC=0.6E0
      SGA=0.63E0
      SGC=.77E0
      RBARA=0.04E0
      RBARC=0.63E0
C     CALCULATE RADII FOR TYPES 8,9,10.
   23 RR(1)=RBARA*EXP(-4.E0*ABS(SGA))
      RR(2)=RBARA
      RR(3)=RBARA*EXP(4.E0*ABS(SGA))
      RR(4)=RBARC*EXP(-4.E0*ABS(SGC))
      RR(5)=RBARC*EXP(4.E0*ABS(SGC))
      MIN=2
      DO 60 J=1,4
      DO 60 I=1,4
      IF (RR(I+1).GT.RR(I)) GO TO 60
      HH=RR(I)
```

```
      RR(I)=RR(I+1)
      RR(I+1)=HH
60    CONTINUE
      GO TO 28
C**   TYPE 11:
C     USER SUPPLIED BIMODAL CASE: FOA AND FOC ARE THE NUMBER DENSITIES
C     FOR THE ACCUMULATION (SMALLER RBAR) AND COARSE MODES,
C     RESPECTIVELY, IN PARTICLES PER CUBIC CENTIMETER.
C     SGA IS STD.DEVIATION FOR MODE A ** NOT LN(SIGMA) ***
C     SGC IS STD.DEVIATION FOR MODE C ** NOT LN(SIGMA) ***
C     *** NOTE, HOWEVER, THAT SGA AND SGC ARE THE LOGS OF THE
C     STANDARD DEVIATIONS IN THE PRE-CODED CASES TYPE  8-10.
   24 READ (IOIN,1) FOA,RBARA,SGA,FOC,RBARC,SGC
      WRITE (IOUT,25) FOA,RBARA,SGA,FOC,RBARC,SGC
      SGA=ABS(ALOG(SGA))
      SGC=ABS(ALOG(SGC))
      GO TO 23
C**   TYPE 12:  MARSHALL-PALMER RAIN MODEL.
C     C.F. MASON, PHYSICS OF CLOUDS, CH. ON RADAR METEOROLOGY.
C     INPUT PARAMETER RAIN IS RAIN RATE IN MILLIMETERS/HOUR;
C     ** EMA,CAYA, AND RHOA ARE REQUIRED FOR THIS DISTRIBUTION.
   26 READ (IOIN,1) RAIN
      ENZERO=0.08E0
      CAPL=41.E0*RAIN**(-0.21E0)
      DENS=ENZERO/CAPL
      AVOL=PI*(CAPL**(-3.E0))*1.E12
C     CONVERT UNITS FROM CM-4 TO (CM-3)*(MICROMETERS**(-1));
C     THE FACTOR OF 2 CONVERTS THE M-P FORMULA FROM DIAMETER-DATA TO
C     RADIUS BASED FORM.
      ENZERO=2.E-4*ENZERO
      CAPL=2.E-4*CAPL
      MIN=0
      RR(1)=1.E-4
      RR(2)=2500.E0
      WRITE (IOUT,27) RAIN,DENS
C     THE NEXT BLOCK IS COMMON TO ALL DISTRIBUTIONS.
C     IT SETS THE NMAX VALUES OF RADIUS, R(KK).
   28 MAX=JDIMCK(2)
      NHALV=MAX-MIN
      NMAX=1+2**MAX
      NI=2**MIN
      IF (NMAX.GT.JDIMCK(3).OR.NI.GT.JDIMCK(3)) CALL DIMER(3)
      NLAST=NI+1
      NKG=2**NHALV
      ENKG= FLOAT(NKG)
      IF (IDSTP.EQ.0) GO TO 30
      DO 29 I=1,NI
      DR(I)=RR(I+1)-RR(I)
      DO 29 K=1,NKG
      KK=(I-1)*NKG+K
   29 R(KK)=RR(I)+( FLOAT(K-1))*DR(I)/ENKG
      R(NMAX)=RR(NLAST)
C     BRANCH AGAIN CALCULATE THE DIFFERENT F(R) ON THE NMAX POINTS R(K)
                         GO TO (31,33,35,38,41,41,39,43,43,43,43,46),IDSTP
C**   TYPE 0: ARBITRARY
C     INTERPOLATE TO EQUAL INCREMENTS OVER RADII
30    DELR=(R(NRADI)-RLO)/ENKG
      F(1)=FF(1)
      NMAXM1=NMAX-1
      DO 64 KK=1,NMAXM1
      RADUS=RLO+DELR*FLOAT(KK)
      DO 62 J=1,NRADI
      K=J
      IF (R(J).GE.RADUS) GO TO 61
62    CONTINUE
61    CONTINUE
      F(KK+1)=(RADUS-R(K-1))*(FF(K)-FF(K-1))/
     1        (R(K)-R(K-1))+FF(K-1)
64    CONTINUE
      DO 65 I=1,NI
```

```
      DR(I)=RR(I+1)-RR(I)
      DO 65 K=1,NKG
      KK=(I-1)*NKG+K
   65 R(KK)=RR(I)+( FLOAT(K-1))*DR(I)/ENKG
      R(NMAX)=RR(NLAST)
      GO TO 48
C**   TYPE 1: LOG NORMAL
   31 DEN=2.E0*SIGMA*SIGMA
      DO 32 KK=1,NMAX
      GNUM=ALOG(R(KK)/RBAR)
   32 F(KK)=EXP(-GNUM*GNUM/DEN)*A/R(KK)
      GO TO 48
C**   TYPE 2: DOUBLE EXPONENTIAL
   33 DO 34 KK=1,NMAX
      FKK=(1.E0-CUE)*B*EXP(-B*R(KK))
   34 F(KK)=FKK+CUE*A*EXP(-A*R(KK))
      GO TO 48
C**   TYPE 3: DEIRMENDJIAN  MODEL C.
   35 DO 36 KK=1,NMAX
   36 F(KK)=1.E0
      NKG1=NKG+1
      DO 37 KK=NKG1,NMAX
   37 F(KK)=(RR(2)/R(KK))**4.E0
      GO TO 48
C**   TYPES 4 AND 7: POWER LAW
   38 GO TO 39
   39 DO 40 KK=1,NMAX
   40 F(KK)=CUE*R(KK)**(-A)
      GO TO 48
C**   TYPE 5 AND TYPE 6:  MODIFIED GAMMA
   41 DO 42 KK=1,NMAX
   42 F(KK)=(EXP(-B*R(KK)**GAM))*R(KK)**ALF
      GO TO 48
C**   TYPES 8,9,10,11:   BIMODAL LOG-NORMAL DISTRIBUTIONS
   43 DENA=2.E0*SGA*SGA
      DENC=2.E0*SGC*SGC
      FAA=FOA/SGA
      FCC=FOC/SGC
      DO 44 KK=1,NMAX
      GNUMA=ALOG(R(KK)/RBARA)
      GNUMC=ALOG(R(KK)/RBARC)
      FA=FAA*EXP(-GNUMA*GNUMA/DENA)
      FC=FCC*EXP(-GNUMC*GNUMC/DENC)
   44 F(KK)=(FA+FC)/R(KK)
      DENS=FOA+FOC
      WRITE (IOUT,45) DENS
      VOLA=4.18879E0*(RBARA**3.E0)*EXP(4.5E0*SGA*SGA)*FOA
      VOLC=4.18879E0*(RBARC**3.E0)*EXP(4.5E0*SGC*SGC)*FOC
      AVOL=(VOLA+VOLC)/DENS
      GO TO 48
C**   TYPE 12: MARSHALL-PALMER RAIN MODEL
   46 DO 47 KK=1,NMAX
   47 F(KK)=ENZERO*EXP(-CAPL*R(KK))
C     CALCULATE NORMALIZED F(KK) AND SOME DRY VOLUMES USING ALL NMAX
C     VALUES OF RADII.
C     (VOL=AVERAGE PARTICLE VOLUME IN A DISTRIBUTION). THE
C     NORMALIZATION AND FURTHER VOLUMES ARE RECALCULATED LATER
C     BY THE HALVING INTEGRATION METHOD.
   48 FSUM=0.E0
      IF(F(1).LT.0.E0)F(1)=0.0E0
      DO 49 J=2,NMAX
      IF(F(J).LT.0.E0)F(J)=0.E0
   49 FSUM=FSUM+0.5E0*(F(J)+F(J-1))*(R(J)-R(J-1))
      DO 50 J=1,NMAX
50    F(J)=F(J)/FSUM
      WRITE (IOUT,8) FSUM
      NRADI =NMAX
      IF (IDSTP.EQ.1.OR.IDSTP.EQ.5.OR.IDSTP.GE.8) WRITE (IOUT,51) AVOL
      VOL=0.E0
      DO 52 J=2,NMAX
```

```
52 VOL=VOL+2.0944E0*(F(J)*R(J)**3.E0+F(J-i)*R(J-1)**3.E0)*(R(J)-R(J-
  1       1))
   WRITE (IOUT,53) VOL
C     THE VOLUME PER PARTICLE CALCULATED HERE IS OBTAINED USING
C     ALL AVAILABLE (NMAX VALUES) VALUES FOR THE PARTICLE RADII.
   WRITE (IOUT,54)
   DO 56 INT=1,NI
   INF=INT+1
56 WRITE (IOUT,55) INT,RR(INT),RR(INF)
 1 FORMAT (6E12.6,I3)
 2 FORMAT (1H ,//24H AEROSOL PARAMETERS ARE )
 3 FORMAT (1H ,24X,6HRBAR= ,E12.6,5X,7HSIGMA= ,E12.6,7H RLO = ,E12.6,
  +          7H RHI = ,E12.6/)
 4 FORMAT (1H ,24X,5HRLO= ,E10.4,1X,5HRHI= ,E10.4,1X,5HCUE= ,E10.4,
  +          1X,3HA= ,E10.4,1X,3HB= ,E10.4/)
 5 FORMAT (1H ,24X,5HRLO= ,E10.4,1X,5HRHI= ,E10.4,1X,5HCUE= ,E10.4,
  +          1X,3HA= ,E10.4,1X,4HVIS=,E10.4/)
 6 FORMAT (1H ,24X,5HRLO= ,E10.4,1X,5HRHI= ,E10.4,1X,4HRC= ,E10.4,1X,
  +          5HALF= ,E10.4,1X,5HGAM= ,E10.4/)
 7 FORMAT (/1H ,5(26H    RADIUS     RELATIVE NO. )//)
 8 FORMAT (/46H NORMALIZATION FACTOR FOR SIZE DISTRIBUTION = ,E14.7)
11 FORMAT (1X,10(1PE12.6,1X))
25 FORMAT (/1X,7HN(A) = ,E12.6,2X,9H RBARA = ,E12.6,2X,12H SIGMA(A) =
  +          ,E12.6,/,1X,7HN(C) = ,E12.6,2X,9H RBARC = ,E12.6,2X,
  +          12H SIGMA(C) = ,E12.6/)
27 FORMAT (/1X,42H MARSHALL-PALMER RAIN MODEL : RAIN RATE = ,1PE10.3,
  +          21H MM PER HOUR, DENS = ,1PE12.6,8H PART/CC)
45 FORMAT (/,1H ,50H*** BIMODAL DISTRIBUTION...EQUIVALENT  DENSITY IS
  +          ,1PE13.6,18H  PARTICLES PER CC,/)
51 FORMAT(/45H AVERAGE ANALYTIC DRY VOLUME PER PARTICLE IS ,3X,
  +          1PE12.6,18H CUBIC MICROMETERS)
53 FORMAT (1X,47HAVERAGE NUMERICAL DRY VOLUME IS                ,
  +          1PE12.6,18H CUBIC MICROMETERS/)
54 FORMAT (1X,10X,35H SIZE-INTERVALS USED ARE AS FOLLOWS/)
55 FORMAT (1H ,14H INTERVAL NO. ,I3,5X,7HRMIN = ,F11.5,5X,8H RMAX =
  +          ,F11.5)
   RETURN
   END
```

```
      SUBROUTINE AGXP2(RELHUM,CTSUM,CSSUM,CRSUM,TVOL,TMASS,DENS,CATTN,
     1 TEMP,DELTA,NINDX,IW,OLSTAR,OM2,LLLL,IT,WAVE,EM,CAY,EMM,MQRTE,
     2 PFNZRO)
      REAL KEXT,KEXTT,KEXOLD
C     N.B. FFF IS AN ALIS FOR ARRAY FF
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FFF(514),
     +NRADI,PI,IDSTP,NKG,NHALV,NI
      COMMON /PT2/ PHH(65),PSUMTT(65),PGG(65),PSUM(65),PSUMT(65),
     1P(65)
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
C     IN THIS SUBROUTINE THE FOLLOWING CONVENTIONS ARE USED IN
C     PREFIXING VARIABLE NAMES:
C     THE LETTER  C  IS USED FOR CROSS-SECTIONS
C     THE LETTER  Q  IS USED FOR EFFICIENCY FACTORS
C     THE LETTER  K  IS USED FOR EXTINCTION COEF. PER UNIT PATH (KM)
C     THE LETTER  T  IS A SUFFIX FOR TOTAL VALUES
C     THE LETTER  O  IS A PREFIX FOR OMEGA SUB 1 AND 2 CALCULATIONS
C     FOR THE IDSTP=6 AND 12 CASES, DENS IS USED TO TRANSFER THE
C     LIQUID WATER CONTENT FROM THE MAIN PROGRAM TO THIS SUBROUTINE:
C     ELWC IS USED AS THE AEROSOL CONCENTRATION FOR THOSE CASES.
      IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) ELWC=DENS
      PZRSMT=0.
      OLSTAR=0.0E0
      OM2=0.E0
      CTSUMT=0.0E+00
      CSSUMT=0.0E+00
      DENST=0.E0
      CRSUMT=0.0E+00
      EMM=1.E0
      NLINES=0
      BH=1.056E-3
C     FACTORS BH AND CH ARE USED IN SIZE ADJUSTMENTS
C     FH IS THE SATURATION RATIO
      FH=RELHUM/100.E0
      CH=FH/(1.E0-FH)
      CONCT=0.0E0
      KEXTT=0.E0
C     CONVERT VOL PER PARTICLE RECEIVED FROM MAIN PROGRAM VIA VARIABLE
C     TVOL TO DRY VOLUME PER PARTICLE IN CUBIC CENTIMETERS
      DRYVOL=TVOL*1.0E-12
      TVOL=0.E0
      TMASS=0.E0
      DO 6 J=1,IT
      PSUMTT(J)=0.0E0
      PHH(J)=0.0E0
    6 PGG(J)=0.E0
C     CONVERT TEMP. TO KELVIN FOR SUBROUTINE WATER USAGE
      TEMK=TEMP+273.16E0
C     SKIP SUBROUTINE WATER FOR THE IDSTP = 12 CASE, AND READ THE
C     OPTICAL DATA FOR RAIN AS EMUA,CAYA,ETC...NEEDED BECAUSE CASE
C     IDSTP=12 MAY BE AT WAVELENGTHS LONGER THAN FOUND IN ROUTINE
C     WATER.
      IF (IDSTP.EQ.12) GO TO 8
C     SUBROUTINE WATER RETURNS INTERPOLATED VALUES FOR EMW, CAYW AND
C     RHOW AT WAVELENGTH = WAVE AND AT TEMPERATURE = TEMK (DEG K).
C     EMW IS REAL PART OF INDEX OF REFR FOR PURE WATER AT TEMP(DEG C).
C     CAYW IS IMAG. PART OF INDEX OF REFR. FOR PURE WATER:
C     CAYW, HERE IS POSITIVE, BUT TREATED AS NEGATIVE IN MIE-ROUTINE.
C     RHOW IS MASS DENSITY(GM/CC) AT TEMPERATURE = TEMP (DEG C).
      CALL WATER(WAVE,EMW,CAYW,TEMK,RHOW)
      WRITE (IOUT,9) EMW,CAYW,TEMP,RHOW
C     BEGIN LOOP OVER AEROSOL COMPONENTS INDEXED BY NK
    8 DO 32 NK=1,NINDX
C     BYPASS READ OF EMA,CAYA,ETC. FOR IDSTP=6 CASE..USE WATER DATA
      IF (IDSTP.NE.6) GO TO 10
      EMA=EMW
      CAYA=CAYW
      RHOA=RHOW
      CONC=ELWC
```

```
      EMUA=0.0E0
      GO TO 11
C     ***  READ OPTICAL AND PHYSICAL DATA   ***
   10 READ (IOIN,2) EMA,CAYA,EMUA,RHOA,CONC
      IF (IW.EQ.0) EMA=EMW
      IF (IW.EQ.0) CAYA=CAYW
      IF(IDSTP.NE.12) GO TO 11
      EMUA=0.0E0
      RHOW=RHOA
      EMW=EMA
      CAYW=CAYA
   11 IF (RHOA.LE.0.E0) RHOA=1.E0
      WRITE (IOUT,3) NK,EMA,CAYA,RHOA,EMUA,CONC
      IF (EMA.LT.1.E-30) GO TO 44
      BHT=BH*(298.E0/TEMK)
      IF (EMUA.LE.0.01) CH=0.0
      BC=BHT*CH
      A=1.E0+((RHOA/RHOW)*EMUA*CH)
      AC=A**(1.E0/3.E0)
C     ADJUST EM,RHO AND CAY PER G. HANEL/ADVANCES IN GEOPHYS/1976
      RHO=RHOW+(RHOA-RHOW)/A
      EM=EMW+(EMA-EMW)/A
      CAY=CAYW+(CAYA-CAYW)/A
      CAY=CAY/EM
C     INITIALIZE QUANTITIES USED TO HOLD RUNNING SUMMATIONS OVER
C     RADII FOR THE CURRENT COMPONENT
      CTSUM=0.E0
      CSSUM=0.E0
      CRSUM=0.0E0
      VOL=0.0E0
      OL1SUM=0.0E0
      OL2SUM=0.E0
      PZRSUM=0.
      DO 13 J=1,IT
   13 PSUM(J)=0.0E0
C     PRINT HEADER IF DETAILED MIE RESULTS ARE TO BE PRINTED
      IF (MORTE.EQ.12345) WRITE (IOUT,5)
C     BEGIN ACTUAL LOOP OVER RADIUS INTERVALS FOR THE CURRENT NK VALUE
C     THIS LOOP IS THE ONE IN WHICH THE MIE CALCULATIONS ARE CALLED
C     INTERVALS ARE INDEXED BY I.  THERE ARE NI SUCH INTERVALS.
      DO 26 I=1,NI
      NRADI=2
      D=RR(I+1)-RR(I)
C     RIT IS THE ADJUSTED RADIUS FOR THE RELATIVE HUMIDITY TO BE USED
C     IN THIS PARTICULAR RUN OR PASS
      RIT=RR(I)*AC -(BC/AC)
      IF (RIT.LT.RR(I).OR.RR(I).LT.0.04E0) RIT=RR(I)
      ALPHA=2.E0*PI*RIT/WAVE
C     ROUTINE MIEGX DOES THE ACTUAL MIE CALCULATIONS.
C     NOTE THAT THE IMAG. PART OF THE REFRACTIVE INDEX (CAY) HAS BEEN
C     NORMALIZED THROUGH DIVISION BY THE REAL PART (EM) BEFORE ITS
C     VALUE IS PASSED TO THE MIE-ROUTINE.
C     MIEGX RETURNS THE EXTINCTION EFFICIENCY FACTOR AS QT
C     MIEGX RETURNS THE SCATTERING EFFICIENCY FACTOR AS QS
C     MIEGX RETURNS THE BACK-SCATTERING (RADAR) EFFIC. FACTOR AS QR
C     MIEGX RETURNS THE AVERAGE INTENSITY (I1+I2)/2 IN THE ARRAY P( )
C     AT ANGLES = ARCCOS(  C( )   ),WHERE C( ) IS SET-UP BY
C     SUBROUTINE GUSET OR ANGLE
C     MIEGX ALSO RETURNS THE 2-ND AND 3-RD LEGENDRE EXPANSION COEF.
C     (OMEGA SUB 1 AND OMEGA SUB 2) AS O1STAR AND O2STAR.
      EMD= (EM)
      CAYD= (CAY)
      ALPHAD= (ALPHA)
      CALL MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,O1STRD,O2STRD,
     +C,IT,PFNZRO)
      EM=(EMD)
      CAY=(CAYD)
      ALPHA=(ALPHAD)
      QT=(QTD)
      QS=(QSD)
```

```
      QR=(QRD)                                                          AGB01410
      O1STAR=(O1STRD)                                                   AGB01420
      O2STAR=(O2STRD)                                                   AGB01430
      KK=1+(I-1)*NKG                                                    AGB01450
      IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I),F(KK),ALPHA,QT,QS,QR AGB01440
      FKK=F(KK)                                                         AGB01460
      FKKA=FKK*PI*RIT**2.E0                                             AGB01470
      VOLHH=4.1888*FKK*RIT**3.E0                                        AGB01480
      OL1HH=O1STAR*FKKA*QT                                              AGB01490
      OL2HH=O2STAR*FKKA*QT                                              AGB01500
      CTHH=QT*FKKA                                                      AGB01510
      CSHH=QS*FKKA                                                      AGB01520
      CRHH=QR*FKKA                                                      AGB01530
      DO 14  J=1,IT                                                     AGB01540
      PHH(J)=P(J)*FKK                                                   AGB01550
14    CONTINUE                                                          AGB01560
      PFNZER=PFNZRO*FKK                                                 AGB01570
      RIT=RR(I+1)*AC-(BC/AC)                                            AGB01580
      IF (RIT.LT.RR(I+1).OR.RR(I+1).LT.0.04E0) RIT=RR(I+1)              AGB01590
      ALPHA=2.E0*PI*RIT/WAVE                                            AGB01600
      EMD= (EM)                                                         AGB01610
      CAYD= (CAY)                                                       AGB01620
      ALPHAD= (ALPHA)                                                   AGB01630
      CALL  MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,O1STRD,O2STRD,          AGB01640
     +C,IT,PFNZRO)                                                      AGB01650
      EM=(EMD)                                                          AGB01660
      CAY=(CAYD)                                                        AGB01670
      ALPHA=(ALPHAD)                                                    AGB01680
      QT=(QTD)                                                          AGB01690
      QS=(QSD)                                                          AGB01700
      QR=(QRD)                                                          AGB01710
      O1STAR=(O1STRD)                                                   AGB01720
      O2STAR=(O2STRD)                                                   AGB01730
      KK1=1+NKG*I                                                       AGB01740
      FKK1=F(KK1)                                                       AGB01750
      IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I+1),FKK1,ALPHA,QT,QS,QRAGB01760
      FKK1A=FKK1*PI*RIT**2.E0                                           AGB01770
      VOLHH=(VOLHH+4.188E0*FKK1*RIT**3.E0)*D*0.5E0                      AGB01780
      OL1HH=(OL1HH+FKK1A*QT*O1STAR)*D*0.5E0                             AGB01790
      OL2HH=(OL2HH+FKK1A*QT*O2STAR)*D*0.5E0                             AGB01800
      CTHH=(CTHH+QT*FKK1A)*D*.5E0                                       AGB01810
      CSHH=(CSHH+QS*FKK1A)*D*.5E0                                       AGB01820
      CRHH=(CRHH+QR*FKK1A)*D*.5E0                                       AGB01830
      DO 15  J=1,IT                                                     AGB01840
      PHH(J)=(PHH(J)+P(J)*FKK1)*D*0.5E0                                 AGB01850
15    CONTINUE                                                          AGB01860
      PFNZER=(PFNZER+PFNZRO*FKK1)*D*0.5                                 AGB01870
      FF=0.5E0*D*(FKK+FKK1)                                             AGB01880
      NT=1                                                              AGB01890
      N=1                                                               AGB01900
   16 NJ=NT                                                             AGB01910
      NT=2*NT                                                           AGB01920
      D=0.5E0*D                                                         AGB01930
      VOLGG=0.0E0                                                       AGB01940
      OL1GG=0.0E0                                                       AGB01950
      OL2GG=0.E0                                                        AGB01960
      CTGG=0.E0                                                         AGB01970
      CSGG=0.E0                                                         AGB01980
      CRGG=0.E0                                                         AGB01990
      FT=0.E0                                                           AGB02000
      DO 17  J=1,IT                                                     AGB02010
   17 PGG(J)=0.E0                                                       AGB02020
      PZRTMP=0.                                                         AGB02030
C     NEXT LOOP HANDLES INTERMEDIATE PARTICLE SIZES..THOSE LYING BETWEEN AGB02040
C     RMIN AND RMAX FOR THE CURRENT INTERVAL WHOSE INDEX IS I.          AGB02050
      DO 19  JG=1,NJ                                                    AGB02060
      KK=1+(I-1)*NKG+(2*JG-1)*(NKG/NT)                                  AGB02070
      RIT=R(KK)*AC  -(BC/AC)                                            AGB02080
      IF (RIT.LT.R(KK).OR.R(KK).LT.0.04E0) RIT=R(KK)                    AGB02090
      ALPHA=2.E0*PI*RIT/WAVE                                            AGB02100
```

```
      EMD= (EM)
      CAYD= (CAY)
      ALPHAD= (ALPHA)
      CALL MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,O1STRD,O2STRD,
     +C,IT,PFNZRO)
      EM=(EMD)
      CAY=(CAYD)
      ALPHA=(ALPHAD)
      QT=(QTD)
      QS=(QSD)
      QR=(QRD)
      O1STAR=(O1STRD)
      O2STAR=(O2STRD)
      IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,R(KK),F(KK),ALPHA,QT,QS,QR
      NRADI=NRADI+1
      FKK=F(KK)
      FKKA=FKK*PI*RIT**2.E0
      VOLGG=4.1888E0*FKK*RIT**3.0E0+VOLGG
      OL1GG=OL1GG+O1STAR*FKKA*QT
      OL2GG=OL2GG+O2STAR*FKKA*QT
      CTGG=CTGG+QT*FKKA
      CSGG=CSGG+QS*FKKA
      CRGG=CRGG+QR*FKKA
      DO 18 J=1,IT
      PGG(J)=PGG(J)+P(J)*FKK
18    CONTINUE
      PZRTMP=PZRTMP+PFNZRO*FKK
   19 FT=FT+FKK
C     ADD RESULTS ACCUMULATED DURING PREVIOUS HALVINGS TO THOSE FOUND
C     FOR THE NEW RADII TREATED WITHIN THE LOOP OVER INDEX JG
      VOLHHT=0.5E0*VOLHH+D*VOLGG
      OL1HHT=0.5E0*OL1HH+D*OL1GG
      OL2HHT=0.5E0*OL2HH+D*OL2GG
      CTHHT=0.5E0*CTHH+D*CTGG
      CSHHT=0.5E0*CSHH+D*CSGG
      CRHHT=0.5E0*CRHH+D*CRGG
      DO 20 J=1,IT
   20 PHH(J)=.5E0*PHH(J)+D*PGG(J)
      PFNZER=.5E0*PFNZER+D*PZRTMP
      FFT=0.5E0*FF+D*FT
      IF (CTHHT.LT.1.E-30) GO TO 22
      DEL=ABS(VOLHHT-VOLHH)/ABS(VOLHHT)
      IF (DEL.LE.DELTA) GO TO 21
      GO TO 22
   21 IF (N.GT.2) GO TO 24
C     DO NOT ALLOW DEL LESS THAN DELTA EXIT UNLESS AT LEAST TWO
C     HAVINGS HAVE BEEN DONE
   22 IF (N.EQ.NHALV) GO TO 24
C     MUST EXIT WHEN NHALV HALVINGS HAVE BEEN DONE EVEN IF THE DELTA
C     CRITERION HAS NOT BEEN SATISFIED..SINCE NO MORE VALUES OF RADII
C     ARE AVAILABLE.
      FF=FFT
      CRHH=CRHHT
      OL1HH=OL1HHT
      OL2HH=OL2HHT
      CSHH=CSHHT
      CTHH=CTHHT
      VOLHH=VOLHHT
      N=N+1
      GO TO 16
24    CONTINUE
      IF(N.EQ.NHALV) WRITE(IOUT,124) I
C     SUM QUANTITIES OVER ALL INTERVALS TREATED UP UNTIL NOW
      CTSUM=CTSUM+CTHHT
      CSSUM=CSSUM+CSHHT
      CRSUM=CRSUM+CRHHT
      VOL=VOL+VOLHHT
      OL1SUM=OL1SUM+OL1HHT
      OL2SUM=OL2SUM+OL2HHT
C     AT THIS POINT, PSUM( ) IS THE RUNNING SUM OF THE AVG. INTENSITY
```

```
C       AS SUMMED OVER SIZES
        DO 25 J=1,IT
     25 PSUM(J)=PSUM(J)+PHH(J)
        PZRSUM=PZRSUM+PFNZER
        WRITE (IOUT,27) NK,I,NRADI,CTHHT
        NLINES=NLINES+NRADI
C        END LOOP OVER HALVING INTERVALS INDEXED BY I
     26 CONTINUE
C       CALCULATE PARTICLE NUMBER DENSITY(NO. PER CC) AS DENSC
        DENSC=CONC/(RHOA*DRYVOL)
C       OVERRIDE CALCULATED VALUE OF DENSC WITH DENS IF LLLL =1
        IF (LLLL.EQ.1) DENSC=DENS
C       RECALCULATE CONC FROM OTHER INPUT DATA IF LLLL=1
        IF (LLLL.EQ.1) CONC=DENS*RHOA*DRYVOL
C       REPLACE DENS BY DENSC FOR LATER USE BY AGXP3
        DENS=DENSC
C       WEIGHT CTSUM,ETC. BY NUMBER DENSITIES (DENSC) FOR THIS COMPONENT
        CTSUM=CTSUM*DENSC
        CSSUM=CSSUM*DENSC
        CRSUM=CRSUM*DENSC
        VOL=VOL*DENSC
        OL1SUM=OL1SUM*DENSC
        OL2SUM=OL2SUM*DENSC
        DO 29 J=1,IT
     29 PSUM(J)=PSUM(J)*DENSC
        PZRSUM=PZRSUM*DENSC
C       NOW, SUM OVER COMPONENTS INDEXED BY NK
C       CONCT IS THE TOTAL DRY-AEROSOL CONCENTRATION IN MG PER CC
        CONCT=CONCT+1.E3*CONC
        DENST=DENST+DENSC
        OLSTAR=OL1SUM+OLSTAR
        OM2=OL2SUM+OM2
        CTSUMT=CTSUMT+CTSUM
C       AT THIS POINT, CTSUMT IS THE TOTAL EXTINCTION CROSS SECTION
C       (IN SQ. MICRONS) AS SUMMED OVER ALL COMPONENTS WHICH
C       HAVE BEEN DEALT WITH THUS FAR
        CSSUMT=CSSUMT+CSSUM
        CRSUMT=CRSUMT+CRSUM
        DO 30 J=1,IT
     30 PSUMTT(J)=PSUM(J)+PSUMTT(J)
        PZRSMT=PZRSMT+PZRSUM
        VOL=VOL*1.E-12
C       TVOL IS THE TOTAL VOLUME (IN CM**3) OCCUPIED BY THE AEROSOL
C       PARTICLES.  TVOL IS NOT ACTUALLY USED IN THIS VERSION OF
C       PROGRAM AGAUS.
        TVOL=VOL+TVOL
        EMASS=VOL*RHO
        TMASS=TMASS+EMASS
        KEXOLD=KEXTT
        KEXTT=CTSUMT*1.E-3
        KEXT=KEXTT-KEXOLD
        WRITE (IOUT,31) NK,VOL,EMASS,KEXT
C       VPF(VOL) IS THE VOLUME PACKING FRACTION: THAT IS, THE FRACTION
C       OF EACH CC OF SPACE WHICH IS FILLED BY AEROSOL MATERIAL BELONGING
C       TO THE CURRENT COMPONENT NK.
C       TMASS IS THE TOTAL MASS OF AEROSOL FOUND IN 1 CC OF SPACE.
C       EMASS IS THE MASS OF AEROSOL MATERIAL PER CC ASSOCIATED WITH
C       THE CURRENT COMPONENT NK.
C       KEXT IS THE EXTINCTION COEF.(PER KM) WHICH IS ASSOCIATED WITH
C       THE CURRENT COMPONENT--AS IF IT ALONE WERE PRESENT.
C       KEXTT IS THE SUM OF THE KEXT'S OVER ALL COMPONENTS.
C       END LOOP OVER AEROSOL COMPONENTS INDEXED BY NK.
        IF (NINDX.GT.1) WRITE (IOUT,42) NK
        IF (MQRTE.EQ.12345) WRITE (IOUT,43)
     32 CONTINUE
        IF (NINDX.GT.1) WRITE (IOUT,33) TMASS,KEXTT
        WRITE (IOUT,34) NLINES
        DENS=DENST
        NRADI=NLINES
C       NOW, PERFORM THE FINAL RENORMALIZATIONS TO OBTAIN CTSUM, ETC.
```

```
C     VALUES REPRESENTATIVE OF A SINGLE AVERAGE PARTICLE.
C     CTSUM BECOMES THE EXTINCTION CROSSECTION IN SQ. MICROMETERS PER
C     AVERAGE PARTICLE. THE OTHER QUANTITIES CARRY SIMILAR MEANINGS.
      DO 35 J=1,IT
   35 PSUM(J)=PSUMTT(J)/DENST
      OLSTAR=OLSTAR/CTSUMT
      OM2=OM2/CTSUMT
      CTSUM=CTSUMT/DENST
      CSSUM=CSSUMT/DENST
      CRSUM=CRSUMT/DENST
C     ...AND CONVERT ANG INTENS AT ZERO DEGREES INTO PHASE FUNCT.
C     WITH NORMALIZATION
      PFNZRO=PZRSMT/DENST*(WAVE*WAVE/(PI*CTSUM*EMM*EMM))
      WRITE(IOUT,36) OLSTAR,OM2,PFNZRO
C     CALCULATE ATTENUATION COEFS. IN SQ.METERS PER MILLIGRAM
      CATTN=CTSUMT*1.E-12/CONCT
      CATTNW=1.E-12*KEXTT/TMASS
      WRITE (IOUT,37) CATTN
      IF(RELHUM.GT.1.0)WRITE (IOUT,38) CATTNW
      GO TO 41
   44 WRITE (IOUT,45) EMA
      STOP
    2 FORMAT (4F10.6,E15.7)
    3 FORMAT (1H ,6H INDX=,I3,4H M= ,F10.6,6H K = -,F10.6,9HI.  MASS ,
     +         10HDENSITY = ,F8.6,
     +         17H GROWTH FACTOR = ,F8.4,9H. CONC = ,1PE12.5,7H  GM/CC/)
    4 FORMAT (1X,F10.5,6(2X,1PE11.5))
    5 FORMAT (//54H R(MICRONS)  DRY  RADIUS     N(R)       MIE  SIZE
     +        ,36HQ (EXT)       Q(SCA)          Q(RADAR)/)
    9 FORMAT (1H ,39HINDEX OF REFRACTION FOR PURE WATER IS: ,F8.6,
     +         3H - ,F8.6,1HI//,1X,25HMASS DENSITY OF WATER AT ,F6.2,
     +           11H DEG C IS: ,F8.2,6H GM/CC,/)
  124 FORMAT(/52H *** CONVERGENCE LEVEL NOT REACHED FOR INTERVAL NO. ,
     +          I3,4H ***/)
   27 FORMAT(1H ,19H FOR COMPONENT NO. ,I3,15H  INTERVAL NO. ,I3,1H ,I4,
     +         43H RADII WERE USED.  CONTRIBUTION TO CTSUM = ,1PE12.6)
   31 FORMAT (1H ,/,20H  FOR COMPONENT NO. ,I2,12H :     VPF = ,1PE12.5,
     +7H PER CC,24H   MASS CONCENTRATION = ,E12.5,21H  GM/CC.  KEXT =
     +         E12.5,7H PER KM)
   33 FORMAT (1H ,/,29H  TOTAL MASS CONCENTRATION = ,1PE12.5,7H GM/CC;,
     +15H  TOTAL KEXT = ,E12.5,7H PER KM)
   34 FORMAT (/1X,32H TOTAL NUMBER OF RADII USED WAS ,I5)
   36 FORMAT (/1X,19H ANALYTIC SOLUTIONS,/,
     +16H  OMEGA SUB 1 = ,1PE14.7/,16H  OMEGA SUB 2 = ,1PE14.7/
     +,16H  PFN AT ZERO = ,1PE14.7,/)
   37 FORMAT (1H ,21H ATTENUATION COEF. = ,1PE12.5,12H  SQ-METERS/,
     +         33HMILLIGRAM OF DRY AEROSOL MATERIAL)
   38 FORMAT (1H ,21H ATTENUATION COEF. = ,1PE12.5,12H  SQ-METERS/,
     +         33HMILLIGRAM OF WET AEROSOL MATERIAL/)
   42 FORMAT (1H /,1X,10X,30(1H*),33H  END OF AEROSOL COMPONENT CYCLE ,
     +         7HNUMBER ,I3,2X,30(1H*)//)
   43 FORMAT (1H )
   45 FORMAT (///,1X,11H***** EMA (,F10.6,20H) IS EITHER ZERO OR ,
     +35HNEGATIVE - PROGRAM TERMINATED *****)
41    RETURN
      END
```

```
      SUBROUTINE AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,
     +WAVE,EM,CAY,EMM,IT,IEND,IANG)
      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
     +,NRADI,PI,IDSTP,NKG,NHALV,NI
C     IEND=1 WHEN THE COMPOSITE PHASE FUNCTION IS BEING WRITTEN
      IF (IEND.EQ.1) GO TO 6
      ALBDO=CSSUM/CTSUM
C     PFACT IS USED TO CONVERT AVG. INTENSITY PSUM() INTO PHASE-
C     FUNCTIONS.  SFACT IS USED TO CONVERT PSUM INTO SCATTERING
C     FRACTIONS, NORMALIZED PER SOM. THE INTEGRAL OF SCAT OVER SOLID
C     ANGLE SHOULD YIELD THE TOTAL SCATTERING CROSS-SECTION IN SQ. M.
      SFACT=WAVE*WAVE*DENS*1.E-6/(4.*PI*PI)
      PFACT=WAVE*WAVE/(PI*CTSUM*EMM*EMM)
      DO 15 J=1,IT
      SCAT=PSUM(J)*SFACT
      PSUM(J)=PSUM(J)*PFACT
C     UNCOMMENT THE FOLLOWING STMT IF U WANT SCAT FRACT,COSINES AND A
C     COUNTER WRITTEN ON NUNIT. NUNIT IS SET IN THE BLOCK DATA ROUTINE.
C     WRITE (NUNIT,3) SCAT,C(J),J
   15 CONTINUE
      IF (NINDX.GE.2) WRITE (IOUT,12)
      WRITE (IOUT,4) IDSTP,WAVE,EM,CAY,CTSUM,CSSUM,ALBDO
C     CONVERT AVG. CROSS-SECTIONS TO COEFFICIENTS (PER KM)
      CTSUM=CTSUM*1.0E-3*DENS
      CSSUM=CSSUM*1.0E-3*DENS
      CRSUM=CRSUM*1.0E-3*DENS
      WRITE (IOUT,13) CTSUM,CSSUM,CRSUM
      WRITE (IOUT,14) GNU,DENS
      IF (IT.LT.2) GO TO 21
C     WRITE PHASE FUNCTION AT SPECIFIED ANGLES
6     WRITE (IOUT,1)
      WRITE (IOUT,5)
C     FIND ANGLES FROM COSINES
      DO 2 I=1,IT
2     FF(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI
      DO 19 J=1,IT,4
      K=J+3
      IF (K.GT.IT) K=IT
   19 WRITE (IOUT,8) (C(I),FF(I),PSUM(I),I=J,K)
      IF (IANG.NE.0) RETURN
C     ROUTINE GAUS GENERATES AND PRINTS THE LEGENDRE
C     EXPANSION COEFS (OMEGAS) FOR THE PHASE FUNCTION.
      CALL GAUS(IT)
C     CHECK TO SEE IF SNG. SCAT. ALBEDO (ALBDO) COMPUTED DIRECTLY
C     FROM CROSS-SECTIONS AGREES WITH THAT FOUND FROM THE LEGENDRE
C     EXPANSION OF PHASE-FUNCTION.
      IF ((ABS(OL(1)-ALBDO)/ALBDO.GT.5.E-3).AND.(IEND.NE.1))
     1         WRITE (IOUT,20)
    1 FORMAT (//1H ,50X,14HPHASE FUNCTION/,1X,42X,31H(NORMALIZED TO 4 PI
     + OMEGA ZERO)//)
3     FORMAT (2(E13.7,1X),I3)
    4 FORMAT (1H1,/,41H  DISTRIBUTION  WAVELENGTH     REFRACTIVE,9X,
     +20HEXTINCTION X SECTION,8X,20HSCATTERING X SECTION,12X,5HALBDO/
     +1H ,6X, 4HTYPE,6X,9H(MICRONS),8X,5HINDEX,16X,12H(SQ MICRONS),13X,
     +          12H(SQ MICRONS)/1H ,I9,4X,F11.4,F10.4,3H(1-,F7.4,2HI),
     +          7X,1PE14.7,11X,1PE14.7,12X,1PE14.7,/)
    5 FORMAT (1H ,3X,4(5H   MU,2X,' ANGLE ',17H PHASE FUNCTION   )/)
    8 FORMAT (1H ,F9.5,F7.2,E12.5,3(3X,F9.5,F7.2,E12.5))
   12 FORMAT (52H THIS IS A MIXED CASE * SUBSEQUENT REFRACTIVE INDEX ,
     +          34HPRINT-OUTS ARE NOT GENERALLY VALID/)
   13 FORMAT (1H ,10H K(EXT) = ,1PE13.7,11H; K(SCA) = ,E13.7,
     +          11H; K(RAD) = ,E13.7,11H ALL PER KM/)
   14 FORMAT (/14H WAVENUMBER = ,1PE12.6,5H CM-1,5X,10HDENSITY = ,E12.6,
     +          17H PARTICLES PER CC/)
   20 FORMAT (//12H *** VALUES
     +          ,55HOF ALBDO AND OL(1) DISAGREE BY MORE THAN 0.5 PERCENT **
     +          ,34HLARGER VALUE OF 'IT' IS NEEDED ***/)
```

```
21    RETURN                                                          AGC00710
      END                                                             AGC00720
```

```
      SUBROUTINE ANGLE(PI,IANG,IT)
C     THIS ROUTINE IS TO BE USED TO REPLACE GUSET FOR THE PURPOSE
C     OF USING AGAUS TO DO PHASE FUNCTION CALCULATIONS AT -IT-
C     ANGLES BETWEEN 0 AND 180 DEGREES, RATHER THAN AT THE G-L
C     QUADRATURE ABSCISSA VALUES.  IT ALSO READS THE INPUT ANGLES
C     IF IANG=2.
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
      RADS=PI/180.
      DEL=180./FLOAT(IT-1)
      IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) GO TO 4
      IF (IANG.EQ.2) GO TO 2
      DO 1 I=1,IT
      W(I)=DEL*FLOAT(I-1)
1     C(I)=COS(W(I)*RADS)
      RETURN
2     READ(5,100) (W(I),I=1,IT)
4     DO 3 I=1,IT
3     C(I)=COS(W(I)*RADS)
      IANG=1
      RETURN
100   FORMAT(16F5.1)
      END
```

```
      FUNCTION GAMMA(X)
C     GAMMA FUNCTION: TAKEN FROM HANDBOOK OF MATHEMATICAL FUNCTIONS,
C     ABRAMOWITZ AND STEGUN, NOV 1964, PP 256-257.  RECURRENCE FORMULA
C     6.1.16, POLYNOMIAL APPROXIMATION 6.1.35.
      DATA A1,A2,A3,A4,A5
     + /-.5748646,.9512363,-.6998588,.4245549,-.1010678/
C     COMPUTER AND GAMMA FUNCTION LIMITS
      IF (X.GT.34..OR.X.LT.0.) GO TO 3
      GSUM=1.
      N=IFIX(X+.00001)
C     FIND Z .LE. 1.
      Z=X-FLOAT(N)
C     CK FOR Z BEING INTEGER
      IF (Z.LT.1.E-04) N=N-1
      IF (Z.LT.1.E-04) Z=1.
C     COMPENSATE FOR N-1 IN FORMULA
      N=N-1
C     IF Z .LE. 1. SKIP LOOP
      IF (N.LE.0) GO TO 2
C     RECURRENCE RELATION: G(N+Z)=(N-1+Z)*(N-2+Z)...(1+Z)*G(1+Z)
      DO 1 I=1,N
      VALUE=FLOAT(I)+Z
1     GSUM=GSUM*VALUE
C     POLYNOMIAL APPROXIMATION: Z.LE.1
2     GAMMA=1.+A1*Z+A2*Z*Z+A3*Z*Z*Z+A4*Z*Z*Z*Z+A5*Z*Z*Z*Z*Z
      GAMMA=GAMMA*GSUM
      RETURN
3     WRITE (1,100) X
100   FORMAT(1H ,' ***** THE VALUE OF X (',2PE11.4,') IS EITHER ',
     +'OUTSIDE COMPUTER LIMITS',/,' OR NEGATIVE - PGM STOPPED *****')
      STOP
      END
```

```
      SUBROUTINE GUSET(IT)
C       THIS ROUTINE CALCULATES THE ABSCISSAE C( )
C       AND GAUSS-LEGENDRE WEIGHTS W( ) FOR NUMERICAL INTEGRATION
C       VIA GAUSS-LEGENDRE QUADRATURE OF ORDER N
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
     +,NRADI,PI,IDSTP,NKG,NHALV,NI
      N=IT
      TOL=1.0E-06
      AA=2.0E+00/PI**2.0E+00
      AB=-62.E+00/(3.0E+00*PI**4.0E+00)
      AC=15116.0E+00/(15.0E+00*PI*6.0E+00)
      AD=-12554474.0E+00/(105.0E+00*PI*8.0E+00)
      F(1)=1.0E+00
      EN= FLOAT(N)
      NP1=N+1
      U=1.0E+00-(2.0E+00/PI)**2.0E+00
      D=1.0E+00/SQRT((EN+0.5E+00)**2.E+00+U/4.0E+00)
      DO 1 I=1,N
      S= FLOAT(I)
      A=4.0E+00*S-1.0E+00
      AE=AA/A
      AF=AB/A**3.0E+00
      AG=AC/A**5.0E+00
      AH=AD/A**7.0E+00
    1 R(I)=PI*(A+AE+AF+AG+AH)/4.E+00
      DO 6 K=1,N
      X=COS(R(K)*D)
    2 F(2)=X
      DO 3 NN=3,NP1
      ENN= FLOAT(NN-1)
      F(NN)=((2.0E+00*ENN-1.E+00)*X*F(NN-1)-(ENN-1.E+00)*F(NN-2))/ENN
3     IF (ABS(F(NN)).GT.1E+35) F(NN)=SIGN(1.E+35,F(NN))
      PNP=EN*(F(N)-X*F(NP1))/(1.0E+00-X*X)
      XI=X-F(NP1)/PNP
      XD= ABS(XI-X)
      XDD=XD-TOL
      IF (XDD) 5,5,4
    4 X=XI
      GO TO 2
    5 C(K)=X
6     W(K)=2.0E+00*(1.0E+00-X*X)/(EN*F(N)*EN*F(N))
      DO 7 I=1,N
      R(I)=0.00
    7 F(I)=0.00
      RETURN
      END
```

```
      SUBROUTINE MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,01STRD,02STRD,
     +C,IT,PFNZRO)
C     THIS ROUTINE IS CURRENTLY SINGLE PRECISION COMPLEX
C     CHANGE THE VALUE OF NDIM IF YOU CHANGE THE DIM OF A IN NEXT LINE
      COMPLEX  A(600),ACAPN,ZNUM,ZDEN,ZPDT,ZRPDT,ZAN,ZANP,Y,RF,RRF,
     1 RRFX,WM1,FNA,FNB,TC1,FNAP,FNBP,FNAPP,FNBPP,TC2,WFN(2)
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
      DIMENSION P(65),C(65)
      DIMENSION T(4),TA(4),TB(2),TC(2),TD(2),TE(2),TF(2),TG(2)
      DIMENSION ELTRMX(4,76),PI(3,76),TAU(3,76)
      EQUIVALENCE (WFN(1),TA(1)),(FNA,TB(1)),(FNB,TC(1)),(FNAP,TD(1))
      EQUIVALENCE (FNBP,TE(1)),(FNAPP,TF(1)),(FNBPP,TG(1))
C      THESE EQUIVALENCES ALLOW USE OF REAL AND IMAG PARTS INDIVIDUALLY
      TOL = 1.E-06
      ITT=IT
      X=(ALPHAD)
      CAY=(CAYD)
      EM=(EMD)
      CAYE=CAY*EM
      QRT=0.0
      S=1.0
      RF=CMPLX(EM,-CAYE)
      NMX=IFIX(X*(EM+CAYE))+9
      RRF=1.0/RF
      RX=1.0/X
      RRFX=RRF*RX
C     LOOP POINT FOR CALCULATING PFN AT ZERO DEGREES
      IAPXCT=0
   21 CONTINUE
C     THESE ARE THE PI AND TAU FUNCTIONS
      DO 1 J=1,IT
      PI(1,J)=0.0
      PI(2,J)=1.0
      TAU(1,J)=0.0
      TAU(2,J)=C(J)
    1 CONTINUE
      T(1)=COS(X)
      T(2)=SIN(X)
      WM1=CMPLX(T(1),-T(2))
      WFN(1)=CMPLX(T(2),T(1))
      WFN(2)=RX*WFN(1)-WM1
      T(1)=CAYE*X
      N=1
C   NDIM MUST EQUAL THE DIMENSION OF A( ).
      NDIM=600
      IF (NMX.LT.NDIM) NDELTA=NMX
      IF (NMX.GT.NDIM) NDELTA=NDIM
      NMX=0
      IF (N.EQ.1) GO TO 4
    2 EN=FLOAT(N)
      T(1)=2.0*EN-1.0
      T(2)=EN-1.0
      T(3)=2.0*EN+1.0
      DO 3 J=1,IT
      PI1J=PI(1,J)
      PI2J=PI(2,J)
      CJ= C(J)
C     SWITCH FOR CALCULATING PFN AT ZERO DEGREES
      IF (IAPXCT.EQ.1) CJ=1.0
      S2T=(1.0-CJ*CJ)
      PI(3,J)=(T(1)*PI2J*CJ-EN*PI1J)/T(2)
      TAU(3,J)=CJ*(PI(3,J)-PI1J)-T(1)*S2T*PI2J+TAU(1,J)
    3 CONTINUE
      WM1=WFN(1)
      WFN(1)=WFN(2)
      WFN(2)=T(1)*RX*WFN(1)-WM1
    4 CONTINUE
C     CALCULATE RATIO OF BESSEL FNS OF CONSECUTIVE ORDER
      IF (N.LT.(NMX+1)) GO TO 9
      NMX=NMX+NDELTA
```

```
      NMIN=NMX+1-NDELTA
      V=FLOAT(NMX)+0.50
      Y=RF*X
      ZANP=2.0/Y
      ZNUM=ZANP*V
      ZPDT=ZNUM
      V=V+1.0
      ZDEN=ZANP*V
      ZNUM=ZDEN-1.0/ZNUM
    5 ZRPDT=ZNUM/ZDEN
      ZPDT=ZRPDT*ZPDT
      IF (ABS(REAL(ZRPDT)-1.0).LT. TOL ) GO TO 7
      IF (V.LT.20000.0) GO TO 6
      WRITE (IOUT,1000) X,EM,CAYE
      STOP
    6 V=V+1.0
      ZAN=ZANP*V
      ZNUM=ZAN-1.0/ZNUM
      ZDEN=ZAN-1.0/ZDEN
      GO TO 5
    7 CONTINUE
      J=NMX
    8 JJ=J-NMX+NDELTA
      A(JJ)=-(FLOAT(J))/Y+ZPDT
      J=J-1
      IF (J.LT.NMIN) GO TO 9
      ZPDT=(2.0*FLOAT(J)+1.0)/Y-1.0/ZPDT
      GO TO 8
    9 CONTINUE
      J=N-NMX+NDELTA
      ACAPN=A(J)
      IF (N.GT.1) GO TO 11
C     THIS PART FOR N EQUAL 1 ONLY
      TC1=ACAPN*RRF+RX
      TC2=ACAPN*RF+RX
C     SEE EQUIVALENCE STMTS FOR EXPANATION OF TA( ), ETC.
      FNA=(TC1*TA(3)-TA(1))/(TC1*WFN(2)-WFN(1))
      FNB=(TC2*TA(3)-TA(1))/(TC2*WFN(2)-WFN(1))
      FNAP=FNA
      FNBP=FNB
      T(1)=1.50
      TB(1)=T(1)*TB(1)
      TB(2)=T(1)*TB(2)
      TC(1)=T(1)*TC(1)
      TC(2)=T(1)*TC(2)
      DO 10 J=1,IT
      TAU2J=TAU(2,J)
      ELTRMX(1,J)=TB(1)+TC(1)*TAU2J
      ELTRMX(2,J)=TB(2)+TC(2)*TAU2J
      ELTRMX(3,J)=TC(1)+TB(1)*TAU2J
      ELTRMX(4,J)=TC(2)+TB(2)*TAU2J
   10 CONTINUE
      QEXT=2.0*(TB(1)+TC(1))
      QSCAT=(TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2))/0.750
      O1STAR=0.0
      O2STAR=0.0
      SUMRR=2.0*(TB(1)-TC(1))
      SUMRI=2.0*(TB(2)-TC(2))
      N=2
      GO TO 2
   11 CONTINUE
      TC1=ACAPN*RRF+EN*RX
      TC2=ACAPN*RF+EN*RX
C     SEE EQUIVALENCE STMTS FOR EXPLANATION OF TA( ), ETC.
      FNA=(TC1*TA(3)-TA(1))/(TC1*WFN(2)-WFN(1))
      FNB=(TC2*TA(3)-TA(1))/(TC2*WFN(2)-WFN(1))
      T(4)=T(1)/(EN*T(2))
      T(2)=(T(2)*(EN+1.0))/EN
      S=-S
      SUMRR=SUMRR+S*T(3)*(TB(1)-TC(1))
```

```
      SUMRI=SUMRI+S*T(3)*(TB(2)-TC(2))
C     SEE LATER COMMENTS ABOUT FOLLOWING STATEMENTS
      QRTL1=QRT
      QRT=SUMRR*SUMRR+SUMRI*SUMRI
C     O1STAR CALCULATION
      O1STAR=O1STAR+(TB(1)*TD(1)+TB(2)*TD(2)+TC(1)*TE(1)+TC(2)*TE(2))
     1 *T(2)*4.0+4.0*T(4)*(TD(1)*TE(1)+TD(2)*TE(2))
      IF (N.LT.3) GO TO 12
C     O2STAR CALCULATION
      F1=TF(1)*TF(1)+TF(2)*TF(2)+TG(1)*TG(1)+TG(2)*TG(2)
      F2=TB(1)*TF(1)+TB(2)*TF(2)+TC(1)*TG(1)+TC(2)*TG(2)
      F3=TD(1)*TG(1)+TD(2)*TG(2)+TE(1)*TF(1)+TE(2)*TF(2)
      ENL1=EN-1.0
      COF1=2.50*((EN-2.0)*ENL1-3.0)*((EN-2.0)*ENL1-3.0)*(2.0*EN-3.0)/
     1    ((EN-2.0)*ENL1*(2.0*EN-1.0)*(2.0*EN-5.0))
      COF2=7.50*(EN-2.0)*(EN+1.0)/(2.0*EN-1.0)
      COF3=15.0/ENL1
      O2STAR=O2STAR+COF1*F1+COF2*F2+COF3*F3
   12 CONTINUE
      QEXT=QEXT+T(3)*(TB(1)+TC(1))
      T(4)=TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2)
      QSCAT=QSCAT+T(3)*T(4)
      T(2)=EN*(EN+1.0)
      T(1)=T(3)/T(2)
      DO 13 J=1,IT
      PI3J=PI(3,J)
      TAU3J=TAU(3,J)
      ELTRMX(1,J)=ELTRMX(1,J)+T(1)*(TB(1)*PI3J+TC(1)*TAU3J)
      ELTRMX(2,J)=ELTRMX(2,J)+T(1)*(TB(2)*PI3J+TC(2)*TAU3J)
      ELTRMX(3,J)=ELTRMX(3,J)+T(1)*(TC(1)*PI3J+TB(1)*TAU3J)
      ELTRMX(4,J)=ELTRMX(4,J)+T(1)*(TC(2)*PI3J+TB(2)*TAU3J)
   13 CONTINUE
      IF (N.LT.5) GO TO 14
      QRTR=ABS((QRT-QRTL1)/QRT)
C     TEST FOR CONVERGENCE ON QEXT, QSCA, AND QRADAR
      IF ((T(4).LT. TOL ).AND.(QRTR.LT. TOL )) GO TO 16
   14 N=N+1
      DO 15 J=1,IT
      PI(1,J)=PI(2,J)
      PI(2,J)=PI(3,J)
      TAU(1,J)=TAU(2,J)
      TAU(2,J)=TAU(3,J)
   15 CONTINUE
      FNAPP=FNAP
      FNBPP=FNBP
      FNAP=FNA
      FNBP=FNB
      GO TO 2
   16 CONTINUE
      DO 18 J=1,IT
      DO 17 I=1,4
      T(I)=ELTRMX(I,J)
   17 CONTINUE
      ELTRMX(1,J)=T(3)*T(3)+T(4)*T(4)
      ELTRMX(2,J)=T(1)*T(1)+T(2)*T(2)
      ELTRMX(3,J)=T(1)*T(3)+T(2)*T(4)
      ELTRMX(4,J)=(T(2)*T(3)-T(4)*T(1))
      PFNZRO= (ELTRMX(1,J)+ELTRMX(2,J))/2.0
      IF (IAPXCT.EQ.1) GO TO 20
      P(J)=PFNZRO
   18 CONTINUE
C     ELTRMX(2,J)   IS THE VERTICAL COMPONENT SCATTERING I1 (EYE1)
C     ELTRMX(1,J)   IS THE HORIZONTAL COMPONENT SCATTERING I2  (EYE2)
C     ELTRMX(3,J)   IS EQUIVALENT TO EYE3
C     ELTRMX(4,J)   IS EQUIVALENT TO -1.0*EYE4
      T(1)=2.0*RX*RX
      SGT=QEXT*T(1)
      SGS=QSCAT*T(1)
      O1STAR=3.0*O1STAR/(X*X*SGT)
      O2STAR=4.0*O2STAR/(X*X*SGT)
```

```
      SGR=(SUMRR*SUMRR+SUMRI*SUMRI)*RX*RX                          MIEGX110
      QTD=(SGT)                                                    MIEGX120
      QSD=(SGS)                                                    MIEGX130
      QRD=(SGR)                                                    MIEGX140
      O1STRD=(O1STAR)                                              MIEGX150
      O2STRD=(O2STAR)                                              MIEGX160
C     LOOP FOR CALCULATION OF PFN AT ZERO DEGREES - FOR GPHASX     MIEGX170
      IAPXCT=1                                                     MIEGX180
      IT=1                                                         MIEGX190
      GO TO 21                                                     MIEGX200
   20 IT=ITT                                                       MIEGX210
      RETURN                                                       MIEGX220
 1000 FORMAT (52H V GT 20000 ERROR IN CONTINUED FRACTIONS MIE ROUTINE, MIEGX230
     1    11H ** ALPHA =,E12.6,6H EM = ,E12.6,7H CAY = ,E12.6/,    MIEGX240
     2 1X,54H IT IS SUGGESTED THAT TOL=1.E-06 FOR SINGLE PRECISION. )  MIEGX250
      END                                                          MIEGX260
```

```
      SUBROUTINE WATER(WAVE,EMT,CAYT,TMCHUR,RHODEN)                     WAT00010
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                              WAT00020
      REAL LAMBDA(169),NSUBR(169),NSUBI(169)                            WAT00030
      DIMENSION TEMP(7),DENS(7)                                         WAT00040
C   THIS PROGRAM SEARCHES HALE AND QUERRY TABLE FOR REFRACTION VS.      WAT00050
C   WAVELENGTH (APPLIED OPTICS, VOL. 12, NO. 3,MARCH 1973,PG 555)       WAT00060
C   AND THE DENSITY VS. TEMPERATURE (HNDBK OF CHEM AND PHYS).           WAT00070
C   IF THE INPUT VALUES (TMCHUR AND WAVE) ARE NOT IN THE TABLES, A      WAT00080
C   LINEAR INTERPOLATION IS COMPUTED. VALUES ACCURATE TO THREE PLACES.  WAT00090
      INTEGER P,POINT,H                                                 WAT00100
C            TABLES FOLLOW FOR 60 LINES                                 WAT00110
      DATA LAMBDA/0.200,0.225,0.250,0.275,0.300,0.325,0.350,0.375,0.400 WAT00120
     1,        0.425,0.450,0.475,0.500,0.525,0.550,0.575,0.600,0.625,0.650 WAT00130
     2,        0.675,0.700,0.725,0.750,0.775,0.800,0.825,0.850,0.875,0.900 WAT00140
     3,        0.925,0.950,0.975,1.000,1.200,1.400,1.600,1.800,2.000,2.200 WAT00150
     4,        2.400,2.600,2.650,2.700,2.750,2.800,2.850,2.900,2.950,3.000 WAT00160
     5,        3.050,3.100,3.150,3.200,3.250,3.300,3.350,3.400,3.450,3.500 WAT00170
     6,        3.600,3.700,3.800,3.900,4.000,4.100,4.200,4.300,4.400,4.500 WAT00180
     7,        4.600,4.700,4.800,4.900,5.000,5.100,5.200,5.300,5.400,5.500 WAT00190
     8,        5.600,5.700,5.800,5.900,6.000,6.100,6.200,6.300,6.400,6.500 WAT00200
     9,        6.600,6.700,6.800,6.900,7.000,7.100,7.200,7.300,7.400,7.500 WAT00210
     X,        7.600,7.700,7.800,7.900,8.000,8.200,8.400,8.600,8.800,9.000 WAT00220
     1,        9.200,9.400,9.600,9.800,10.00,10.50,11.00,11.50,12.00,12.50 WAT00230
     2,        13.00,13.50,14.00,14.50,15.00,15.50,16.00,16.50,17.00,17.50 WAT00240
     3,        18.00,18.50,19.00,19.50,20.00,21.00,22.00,23.00,24.00,25.00 WAT00250
     4,        26.00,27.00,28.00,29.00,30.00,32.00,34.00,36.00,38.00,40.00 WAT00260
     5,        42.00,44.00,46.00,48.00,50.00,60.00,70.00,80.00,90.00,100.0 WAT00270
     6,        110.0,120.0,130.0,140.0,150.0,160.0,170.0,180.0,190.0,200.0 WAT00280
     7         /                                                        WAT00290
      DATA NSUBR/1.396,1.373,1.362,1.354,1.349,1.346,1.343,1.341,1.339  WAT00300
     1,       1.338,1.337,1.336,1.335,1.334,1.333,1.333,1.332,1.332,1.331  WAT00310
     2,       1.331,1.331,1.330,1.330,1.330,1.329,1.329,1.329,1.328,1.328  WAT00320
     3,       1.328,1.327,1.327,1.327,1.324,1.321,1.317,1.312,1.306,1.296  WAT00330
     4,       1.279,1.242,1.219,1.188,1.157,1.142,1.149,1.201,1.292,1.371  WAT00340
     5,       1.426,1.467,1.483,1.478,1.467,1.450,1.432,1.420,1.410,1.400  WAT00350
     6,       1.385,1.374,1.364,1.357,1.351,1.346,1.342,1.338,1.334,1.332  WAT00360
     7,       1.330,1.330,1.330,1.328,1.325,1.322,1.317,1.312,1.305,1.298  WAT00370
     8,       1.289,1.277,1.262,1.248,1.265,1.319,1.363,1.357,1.347,1.339  WAT00380
     9,       1.334,1.329,1.324,1.321,1.317,1.314,1.312,1.309,1.307,1.304  WAT00390
     X,       1.302,1.299,1.297,1.294,1.291,1.286,1.281,1.275,1.269,1.262  WAT00400
     1,       1.255,1.247,1.239,1.229,1.218,1.185,1.153,1.126,1.111,1.123  WAT00410
     2,       1.146,1.177,1.210,1.241,1.270,1.297,1.325,1.351,1.376,1.401  WAT00420
     3,       1.423,1.443,1.461,1.476,1.480,1.487,1.500,1.511,1.521,1.531  WAT00430
     4,       1.539,1.545,1.549,1.551,1.551,1.546,1.536,1.527,1.522,1.519  WAT00440
     5,       1.522,1.530,1.541,1.555,1.587,1.703,1.821,1.886,1.924,1.957  WAT00450
     6,       1.966,2.004,2.036,2.056,2.069,2.081,2.094,2.107,2.119,2.130  WAT00460
     7        /                                                         WAT00470
C     DATA NSUBI/1.10E-7,4.90E-8,3.35E-8,2.35E-8,1.60E-8,1.08E-8,6.50E-9WAT00480
C    1, 3.50E-9,1.86E-9,1.30E-9,1.02E-9,9.35E-10,1.00E-9,1.32E-9,1.96E-9WAT00490
C    2,  3.60E-9,1.09E-8,1.39E-8,1.64E-8,2.23E-8,3.35E-8,9.15E-8,1.56E-7WAT00500
C    3,  1.48E-7,1.25E-7,1.82E-7,2.93E-7,3.91E-7,4.86E-7,1.06E-6,2.93E-6WAT00510
C    4,  3.48E-6,2.89E-6,9.89E-6,1.38E-4,8.55E-5,1.15E-4,1.10E-3,2.89E-4WAT00520
C    5,  9.56E-4,3.17E-3,6.70E-3,1.90E-2,5.90E-2,1.15E-1,1.85E-1,2.68E-1WAT00530
C    6,  2.98E-1,2.72E-1,2.40E-1,1.92E-1,1.35E-1,9.24E-2,6.10E-2,3.68E-2WAT00540
C    7,  2.61E-2,1.95E-2,1.32E-2,9.40E-3,5.15E-3,3.60E-3,3.40E-3,3.80E-3WAT00550
C    8,  4.60E-3,5.62E-3,6.88E-3,8.45E-3,1.03E-2,1.34E-2,1.47E-2,1.57E-2WAT00560
C    9,  1.50E-2,1.37E-2,1.24E-2,1.11E-2,1.01E-2,9.80E-3,1.03E-2,1.16E-2WAT00570
C    X,  1.42E-2,2.03E-2,3.30E-2,6.22E-2,1.07E-1,1.31E-1,8.80E-2,5.70E-2WAT00580
C    1,  4.49E-2,3.92E-2,3.56E-2,3.37E-2,3.27E-2,3.22E-2,3.20E-2,3.20E-2WAT00590
C    2,  3.21E-2,3.22E-2,3.24E-2,3.26E-2,3.28E-2,3.31E-2,3.35E-2,3.39E-2WAT00600
C    3,  3.43E-2,3.51E-2,3.61E-2,3.72E-2,3.85E-2,3.99E-2,4.15E-2,4.33E-2WAT00610
C    4,  4.54E-2,4.79E-2,5.08E-2,6.62E-2,9.68E-2,1.42E-1,1.99E-1,2.59E-1WAT00620
C    5,  3.05E-1,3.43E-1,3.70E-1,3.88E-1,4.02E-1,4.14E-1,4.22E-1,4.28E-1WAT00630
C    6,  4.29E-1,4.29E-1,4.26E-1,4.21E-1,4.14E-1,4.04E-1,3.93E-1,3.82E-1WAT00640
C    7,  3.73E-1,3.67E-1,3.61E-1,3.56E-1,3.50E-1,3.44E-1,3.38E-1,3.33E-1WAT00650
C    8,  3.28E-1,3.24E-1,3.29E-1,3.43E-1,3.61E-1,3.85E-1,4.09E-1,4.36E-1WAT00660
C    9,  4.62E-1,4.88E-1,5.14E-1,5.87E-1,5.76E-1,5.47E-1,5.36E-1,5.32E-1WAT00670
C    X,  5.31E-1,5.26E-1,5.14E-1,5.00E-1,4.95E-1,4.96E-1,4.97E-1,4.99E-1WAT00680
C    1,  5.01E-1,5.04E-1/                                               WAT00690
C    ALTERNATE FORM OF ABOVE DATA STMT DUE TO EXCESS CONTINUATION CARDS WAT00700
```

```
      DATA (NSUBI(I),I=1,143)
     +              /1.10E-7,4.90E-8,3.35E-8,2.35E-8,1.60E-8,1.08E-8,6.50E-9
     1, 3.50E-9,1.86E-9,1.30E-9,1.02E-9,9.35E-10,1.00E-9,1.32E-9,1.96E-9
     2,   3.60E-9,1.09E-8,1.39E-8,1.64E-8,2.23E-8,3.35E-8,9.15E-8,1.56E-7
     3,   1.48E-7,1.25E-7,1.82E-7,2.93E-7,3.91E-7,4.86E-7,1.06E-6,2.93E-6
     4,   3.48E-6,2.89E-6,9.89E-6,1.38E-4,8.55E-5,1.15E-4,1.10E-3,2.89E-4
     5,   9.56E-4,3.17E-3,6.70E-3,1.90E-2,5.90E-2,1.15E-1,1.85E-1,2.68E-1
     6,   2.98E-1,2.72E-1,2.40E-1,1.92E-1,1.35E-1,9.24E-2,6.10E-2,3.68E-2
     7,   2.61E-2,1.95E-2,1.32E-2,9.40E-3,5.15E-3,3.60E-3,3.40E-3,3.80E-3
     8,   4.60E-3,5.62E-3,6.88E-3,8.45E-3,1.03E-2,1.34E-2,1.47E-2,1.57E-2
     9,   1.50E-2,1.37E-2,1.24E-2,1.11E-2,1.01E-2,9.80E-3,1.03E-2,1.16E-2
     X,   1.42E-2,2.03E-2,3.30E-2,6.22E-2,1.07E-1,1.31E-1,8.80E-2,5.70E-2
     1,   4.49E-2,3.92E-2,3.56E-2,3.37E-2,3.27E-2,3.22E-2,3.20E-2,3.20E-2
     2,   3.21E-2,3.22E-2,3.24E-2,3.26E-2,3.28E-2,3.31E-2,3.35E-2,3.39E-2
     3,   3.43E-2,3.51E-2,3.61E-2,3.72E-2,3.85E-2,3.99E-2,4.15E-2,4.33E-2
     4,   4.54E-2,4.79E-2,5.08E-2,6.62E-2,9.68E-2,1.42E-1,1.99E-1,2.59E-1
     5,   3.05E-1,3.43E-1,3.70E-1,3.88E-1,4.02E-1,4.14E-1,4.22E-1,4.28E-1
     6,   4.29E-1,4.29E-1,4.26E-1,4.21E-1,4.14E-1,4.04E-1,3.93E-1,3.82E-1
     7, 3.73E-1,3.67E-1,3.61E-1,3.56E-1,3.50E-1,3.44E-1,3.38E-1,3.33E-1/
      DATA (NSUBI(I),I=144,169) /
     8    3.28E-1,3.24E-1,3.29E-1,3.43E-1,3.61E-1,3.85E-1,4.09E-1,4.36E-1
     9,   4.62E-1,4.88E-1,5.14E-1,5.87E-1,5.76E-1,5.47E-1,5.36E-1,5.32E-1
     X,   5.31E-1,5.26E-1,5.14E-1,5.00E-1,4.95E-1,4.96E-1,4.97E-1,4.99E-1
     1,   5.01E-1,5.04E-1/
      DATA TEMP/273.,278.,283.,288.,293.,298.,303./,DENS/
     1       0.999841,0.999965,0.999700,0.999099,0.998203,0.997044,
     2       0.995994/
      EMT=0.0
      CAYT=0.0
      POINT=0
      H=0
      IF (WAVE.LT.0.2.OR.WAVE.GT.200.0.OR.TMCHUR.LT.273.OR.
     1     TMCHUR.GT.303.0) GO TO 11
C    BINARY SEARCH
      L=1
      H=125
    1 POINT=((L+H)/2)
      TEST=ABS(LAMBDA(POINT)-WAVE)
      IF (TEST.LE.0.0001) GO TO 4
      IF (WAVE.GT.LAMBDA(POINT)) GO TO 2
      H=POINT
      GO TO 3
    2 L=POINT+1
    3 IF (L.NE.H) GO TO 1
      L=L-1
C    INTERPOLATION ROUTINE
      EMT=NSUBR(L)+(NSUBR(L+1)-NSUBR(L))*((WAVE-LAMBDA(L))/(LAMBDA(L+1)-
     1         LAMBDA(L)))
      CAYT=NSUBI(L)+(NSUBI(L+1)-NSUBI(L))*((WAVE-LAMBDA(L))/(LAMBDA(L+1)
     1         -LAMBDA(L)))
      GO TO 5
    4 CONTINUE
      EMT=NSUBR(POINT)
      CAYT=NSUBI(POINT)
C   SEARCH TEMP VS DENS
    5 IF (TMCHUR.LT.273.0.OR.TMCHUR.GT.303.0) GO TO 11
      L=1
      H=7
    6 P=((L+H)/2)
      TESTT=ABS(TEMP(P)-TMCHUR)
      IF (TESTT.LE.0.0001) GO TO 9
      IF (TMCHUR.GT.TEMP(P)) GO TO 7
      H=P
      GO TO 8
    7 L=P+1
    8 IF (L.NE.H) GO TO 6
      L=L-1
      RHODEN=DENS(L)+(DENS(L+1)-DENS(L))*(TMCHUR-TEMP(L))/(TEMP(L+1)-
     1         TEMP(L))
      GO TO 10
```

```
 9 RHODEN=DENS(P)
10 CONTINUE
   GO TO 13
11 WRITE (IOUT,12) TMCHUR,WAVE
12 FORMAT (10H TEMP. OF ,E12.6,14H OR WAVEL. OF ,E12.6,14H BEYOND RAN
  1GE ,24HOF DATA IN WATER ROUTINE/22H EXECUTION TERMINATED )
   STOP
13 RETURN
   END
```

```
      SUBROUTINE GAUS(IT)
      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
      WRITE (IOUT,11)
C       INITIALIZE ARRAY PC( ) USED FOR RUNNING SUMMATION
      DO 1 I=1,IT
      OL(I)=0.
    1 PC(I)=0.E+00
C     LOOPS  2 AND 3 CALCULATE EXPANSION COEFS. FOR FUNCTION PSUM(  )
C     VIA GAUSS-LEGENDRE QUADRATURE. THE COEFS. GO INTO ARRAY OL( )
      DO 3 I=1,IT
      COF=W(I)*PSUM(I)
      PLM1=C(I)
      PLM2=1.
      DO 2 LL=1,IT
      L=LL-1
      PL=PLM2
      IF (LL.EQ.2) PL=PLM1
      IF (LL.LE.2) GO TO 2
      PL=2.*C(I)*PLM1-PLM2-(C(I)*PLM1-PLM2)/FLOAT(L)
      PLM2=PLM1
      PLM1=PL
2     OL(LL)=OL(LL)+COF*PL*(FLOAT(L)+.5)
3     CONTINUE
      DO 7 I=1,IT
      II=I-1
7     WRITE (IOUT,8) II,OL(I)
    8 FORMAT (1H ,20X,I6,20X,1PE14.7)
11    FORMAT (1H /,1X,25X,1HL,20X,16HL-TH COEFFICIENT)
      RETURN
      END
```

```
      SUBROUTINE DIMER(NGO)
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
      GO TO (1,2,3),NGO
1     WRITE(IOUT,101)
101   FORMAT(1H ,'**** THE INPUT VALUE OF IT IS GREATER '
     1,'THAN THE ARRAY DIMENSIONS',/,'CHANGE THE DIMENSIONS OF THE ',
     2'FOLLOWING ARRAYS IN SUBS AND COMMON',/,'COMMON BLOCK',10X,
     3'ARRAY(S)',/,'     AGXM',10X,'    C,H,OLT',/,'     PT2',11X,
     4'PC,OL,RMS,PSUM,PSUMT,P',/,'ALSO CHANGE THE VALUE OF JIMCK(1) '
     5,'IN THE DATA STMT TO AGREE WITH THE NEW',/,' DIMENSION LIMITS',
     6/,1H ,55HTHE SECOND INDEX ON ARRAYS PI, TAU, AND ELTRMX IN MIEGX
     7,1H ,56HMUST ALSO BE CHANGED AND ARRAYS P AND C CHANGED AS WELL.)
      STOP
2     WRITE (IOUT,102)
102   FORMAT(1H ,'**** TOO MANY PARTICLE RADII FOR DIMENSION LIMITS:',
     1'IN SUBS AND COMMON CHANGE THE FOLLOWING ',
     2'ARRAYS',/,'COMMON BLOCK',10X,'ARRAY(S)',/,'     PT1',11X,'F,R,FF',
     3/,'ARRAYS F AND R MUST BE CONSISTENT WITH THE FOLLOWING; ARRAY ',
     4'SIZE = 1 + 2**JDIMCK(2)',/,'ARRAY FF MUST BE DIMENSIONED TO ONE',
     5' MORE THAN ARRAYS F AND R',/,'ALSO CHANGE THE VALUE OF JDIMCK(2)'
     6,' IN THE DATA STATEMENT')
      STOP
3     WRITE (IOUT,103)
103   FORMAT (1H ,' THE DIMENSIONS OF F AND'
     1,'R DO NOT AGREE WITH THE FOLLOWING:',/,'SIZE=1+2**JDIMCK(2)   ',
     2'WHERE JDIMCK(2) APPEARS IN THE DATA STATEMENT',/,'ALSO DIMENSION'
     3,' ARRAY FF TO BE ONE MORE THAN ARRAYS F AND R')
      STOP
      END
```

```
      BLOCK DATA                                                        BLK00010
C     IF YOU CHANGE THE DIMENSIONS MAKE SURE THAT YOU ALSO CHANGE THE   BLK00020
C     DATA STATEMENT CONTAINING JDIMCK( ):                              BLK0003
C     JDIMCK(1)=ORDER OF QUADRATURE (INPUT IT),I.E. MAX SIZE OF DIMENSIOBLK00040
C     JDIMCK(2):USED IN COMPUTATION OF SIZE OF ARRAYS F AND R,          BLK00050
C     SIZE=1+2**JDIMCK(2), ARRAY FF SHOULD BE ONE MORE THAN F AND R.    BLK00060
C     JIMCK(3) WILL BE CALCULATED.                                      BLK00070
C     THE OTHER ARRAYS ARE REUSED, SO                                   BLK00080
C     CHECK THE SUBROUTINE CALLS AND COMMON BLOCKS TO SEE IF            BLK00090
C     ARRAYS HAVE BEEN RENAMED WHEN U CHANGE DIMENSIONS.                BLK00100
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                       BLK00110
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)                    BLK00120
     +,NRADI,PI,IDSTP,NKG,NHALV,NI                                      BLK00130
      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)       BLK00140
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                              BLK00150
      DATA IOIN,IOUT,NUNIT,IEO,NEOU /5,6,3*0/                           BLK00160
      DATA JDIMCK /65,9,0/                                              BLK00170
C     EOSAEL OPTION:  65 PREDETERMINED ANGLES                           BLK00180
      DATA W                                                            BLK00190
     +   /  0.0,  0.5,  1.0,  2.0,  3.0,  4.0,  5.0,  6.0,  8.0, 10.0,  BLK00200
     +     12.0, 14.0, 16.0, 18.0, 20.0, 24.0, 28.0, 32.0, 36.0, 40.0,  BLK00210
     +     44.0, 48.0, 52.0, 56.0, 60.0, 64.0, 68.0, 72.0, 76.0, 80.0,  BLK00220
     +     84.0, 88.0, 92.0, 96.0,100.0,104.0,108.0,112.0,116.0,120.0,  BLK00230
     +    124.0,128.0,132.0,136.0,140.0,142.0,144.0,146.0,148.0,150.0,  BLK00240
     +    152.0,154.0,156.0,158.0,160.0,162.0,164.0,166.0,168.0,170.0,  BLK00250
     +    172.0,174.0,176.0,178.0,180.0/                                BLK00260
      END                                                               BLK00270
```

```
      PROGRAM FLASH1
      COMMON/IOUNT/IOIN,IOOUT
      COMMON/CONST/PI
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
     *             TIME,ISRC,RCTSEE,RCTLOC
      PI=3.14159
      IOIN=5
      IOOUT=6
C**** NOTE ICHK=-1 DEFAULTS USETR INPUT TO WAVE1
      ICHK=0
      WAVE1=.55
      CALL FLASH(WAVE1,ICHK)
      WRITE (IOOUT,100) WAVE1,ICHK
100   FORMAT (5X,13HEOSAEL OUTPUT,/,
     *5X,13HWAVELENGTH = ,F6.1,1X,11HMICROMETERS,/,
     *5X,7HICHK = ,I4)
      STOP
      END
```

```
      SUBROUTINE FLASH(WAVE1,ICHK)
C***********************************************************************
C*                       PROGRAM FLASH                                 *
C*                         EOSAEL80                                    *
C***********************************************************************
      DIMENSION FLASH1(3),FLASH2(3),IC1(3),IC2(3),IGUN(3)
      COMMON/IOUNT/IOIN,IOOUT
      COMMON/CONST/PI
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
     *             TIME,ISRC,RCTSEE,RCTLOC
      DATA FLASH1/3.85,2.55,0.0/
      DATA FLASH2/0.15,0.15,0.0/
      DATA IC1/2HHO,2HT-,2HUN/
      DATA IC2/2HW.,2H55,2HKN/
      DATA IGUN/105,100,0/
   98 FORMAT(//,21X,30H*****PROGRAM FLASH OUTPUT*****,/,21X,40(2H--))
   99 FORMAT(21X,20H*****END OF RUN*****,/,21X,40(2H--))
  100 FORMAT(1H1,20X,40(2H**),/,21X,1H*,34X,13HPROGRAM FLASH,31X,1H*,/,
     *21X,1H*,37X,8HEOSAEL80,33X,1H*,/,21X,40(2H**))
  101 FORMAT(//,21X,15H*****INPUT*****,/,21X,40(2H--))
  102 FORMAT(21X,14HSCENARIO DATA:,/,21X,14HREFERENCE TIME,1X,F8.3,
     *1X,3HSEC)
  103 FORMAT(21X,9HOBSERVER:,18X,7HTARGET:,20X,7HSOURCE:)
  104 FORMAT(24X,6HX(OBS),3X,F6.1,2H M,10X,6HX(TAR),3X,F6.1,2H M,10X,
     *6HX(SRC),3X,F6.1,2H M,/,
     *          24X,6HY(OBS),3X,F6.1,2H M,10X,6HY(TAR),3X,F6.1,2H M,10X,
     *6HY(SRC),3X,F6.1,2H M,/,
     *          24X,6HZ(OBS),3X,F6.1,2H M,10X,6HZ(TAR),3X,F6.1,2H M,10X,
     *6HZ(SRC),3X,F6.1,2H M)
  105 FORMAT(48X,11HORIENTATION,1X,F6.1,1X,3HDEG,8X,
     *10HEVENT TIME,1X,F6.3,1X,3HSEC,/,67X,
     *10HCCW X-AXIS)
  106 FORMAT(21X,25HDETECTOR CHARACTERISTICS:,/,
     *          24X,13HFIELD OF VIEW,12X,F6.1,1X,7HDEGREES,/,
     *          24X,10HWAVELENGTH,15X,F6.1,1X,11HMICROMETERS,/,
     *          24X,21HRESOLUTION CRITERIA--,/,
     *          26X,17H(A) FOR DETECTION,6X,F6.3,1X,12HMILLIRADIANS,/,
     *          26X,15H(B) FOR LOCK ON,8X,F6.3,1X,12HMILLIRADIANS,/,
     *          24X,24HRECOVERY TIME (R=100M)--,/,
     *          26X,17H(A) FOR DETECTION,6X,F6.1,1X,7HSECONDS,/,
     *          26X,15H(B) FOR LOCK ON,8X,F6.1,1X,7HSECONDS)
  107 FORMAT(21X,23HTARGET CHARACTERISTICS:,/,
     *          24X,6HLENGTH,19X,F6.1,1X,6HMETERS,/,
     *          24X,5HWIDTH,20X,F6.1,1X,6HMETERS,/,
     *          24X,6HHEIGHT,19X,F6.1,1X,6HMETERS,/,
     *          24X,19HEXPOSURE CRITERIA--,/,
     *          26X,17H(A) FOR DETECTION,6X,F6.1,1X,7HPERCENT,/,
     *          26X,15H(B) FOR LOCK ON,8X,F6.1,1X,7HPERCENT)
  108 FORMAT(21X,23HSOURCE CHARACTERISTICS:,/,
     *          24X,4HTYPE,22X,I4,2HMM,1X,2A2,/,
     *          24X,22HFLASH (VISIBLE) RADIUS,4X,F6.3,1X,6HMETERS,/,
     *          24X,24HFLASH (VISIBLE) DURATION,2X,F6.3,1X,7HSECONDS)
  109 FORMAT(24X,27H*****PROGRAM FLASH END*****,/,21X,40(2H--),1H1)
      IWRIT=1
      IFLAG=0
C*****READ IN DATA
    1 WRITE(IOOUT,100)
      CALL DATRD(IWRIT,IFLAG)
      IF(ICHK.EQ.-1)WAVE=WAVE1
      WAVE1=WAVE
      IF(IFLAG.EQ.4)GO TO 9999
      JSRC=ISRC
      IF(ISRC.LT.1.OR.ISRC.GT.2)JSRC=3
      FLASHR=FLASH1(JSRC)
      FLASHT=FLASH2(JSRC)
      WRITE(IOOUT,101)
      WRITE(IOOUT,102)TIME
      WRITE(IOOUT,103)
      WRITE(IOOUT,104)(ROBS(I),RTAR(I),RSRC(I),I=1,3)
```

```
      WRITE(IOOUT,105)TARDEG,TIMGUN                                     FLASH710
      WRITE(IOOUT,106)FOV,WAVE,RADSEE,RADLOC,RCTSEE,RCTLOC              FLASH720
      WRITE(IOOUT,107)TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC                FLASH730
      WRITE(IOOUT,108)IGUN(JSRC),IC1(JSRC),IC2(JSRC),FLASHR,FLASHT      FLASH740
      CALL GETIM(FLASHR,FLASHT,ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)           FLASH750
      WRITE(IOOUT,98)                                                   FLASH760
      CALL DATWT(ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)                         FLASH770
      WRITE(IOOUT,99)                                                   FLASH780
      GO TO 1                                                           FLASH790
 9999 WRITE(IOOUT,109)                                                  FLASH800
C*****DEFINE EOSAEL OUTPUT                                              FLASH810
      ICHK=ISEE+1                                                       FLASH820
      STOP                                                              FLASH830
      END                                                               FLASH840
```

```
      SUBROUTINE DATRD(IWRIT,IFLAG)
C/*********************************************************************/
C/*                       SUBROUTINE DATRD                            */
C/*                         FLASH MODULE                              */
C/*                           EOSAEL80                                */
C/*********************************************************************/
C*****THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS
C     THE SMOKE(EOSAEL) AND GRNAD(EOSAEL) MODULES
C     INPUTS
C       EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,
C       FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.
C       PER FIELD BEGINNING IN COL 11.  THE CARDS ARE NOT ORDER
C       DEPENDENT.
C       NAME                IGNORED
C       SCEN
C                 TIME               SCENARIO REFERENCE TIME(SEC)
C                 ISRC               SOURCE TYPE CODE (1=105MM 2=100MM)
C       OBSC
C                 ROBS(3)            OBSERVER COORDINATES (X,Y,Z METERS)
C       TARC
C                 RTAR(3)            TARGET COORDINATES (X,Y,Z METERS)
C                 TARDEG             TARGET ORIENTATION COUNTERCLOCKWISE TO
C                                    POSITIVE X AXIS
C       SRCC
C                 RSRC(3)            CORDINATES OF FLASH CENTER (X,Y,Z METERS)
C                 TIMGUN             TIME OF INIATION OF GUNFLASH
C       DCHR
C                 FOV                DETECTOR FIELD OF VIEW (DEGREES)
C                 WAVE               DETECTOR WAVELENGTH (MICROMETERS)
C                 RADSEE             ANGULAR RESOLUTION NEEDED TO DETECT
C                                    (MILLIRADIANS)
C                 RADLOC             ANGULAR RESOLUTION NEEDED TO LOCK ON
C                                    (MILLIRADIANS)
C                 RCTSEE             RECOVERY TIME AT 100 METERS FOR
C                                    DETECTION (SECONDS)
C                 RCTLOC             RECOVERY TIME AT 100 METERS FOR
C                                    LOCK ON (SECONDS)
C       TCHR
C                 TARLEN             TARGET LENGTH (METERS)
C                 TARWID             TARGET WIDTH (METERS)
C                 TARHGT             TARGET HEIGHT (METERS)
C                 PCTSEE             FRACTION OF EXPOSURE NEEDED FOR DETECTION
C                                    (PERCENT)
C                 PCTLOC             FRACTION OF EXPOSURE NEEDED FOR LOCK ON
C                                    (PERCENT)
C       GO             SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT
C       DONE           END OF JOB.
C****************************************************************
      COMMON/IOUNT/IOIN,IOOUT
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
     *             TIME,ISRC,RCTSEE,RCTLOC
      DIMENSION IR(18),IR1(2),R1(7),INAME(35)
      DATA IR/2HNA,2HME,2HSC,2HEN,2HOB,2HSC,2HTA,2HRC,2HSR,2HCC,
     *        2HDC,2HHR,2HTC,2HHR,2HGO,2H  ,2HDO,2HNE/
  100 FORMAT(21X,20H*****CARD INPUT*****,/,21X,40(2H--))
  101 FORMAT(2A2,6X,7F10.3)
  102 FORMAT(21X,2A2,6X,7F10.3)
  103 FORMAT(2A2,6X,35A2)
  104 FORMAT(21X,2A2,6X,35A2)
C*****
C***** BEGINNING OF READ LOOP
C*****
      IF(IWRIT.EQ.0)GO TO 6
      WRITE(IOOUT,100)
    6 DO 10 I=1,9
      IF(I.EQ.9)GO TO 90
      IF(IFLAG.GT.0)GO TO 4
      IFLAG=1
      READ(IOIN,103)IR1(1),IR1(2),(INAME(J),J=1,35)
```

```
      IF(IWRIT.EQ.0)GO TO 4
      WRITE(IOOUT,104)IR1(1),IR1(2),(INAME(J),J=1,35)
    4 READ(IOIN,101) IR1(1),IR1(2),(R1(J),J=1,7)
      IF(IWRIT.EQ.0) GO TO 5
      WRITE(IOOUT,102) IR1(1),IR1(2),(R1(J),J=1,7)
    5 IF(IR1(1).EQ.IR(17).AND.IR1(2).EQ.IR(18)) GO TO 998
C*****
C***  RELATING INPUT DATA TO VARIABLE NAMES.
C*****
              IF(IR1(1).EQ.IR(1).AND.IR1(2).EQ.IR(2))    GO TO 10
              IF(IR1(1).EQ.IR(3).AND.IR1(2).EQ.IR(4))    GO TO 20
              IF(IR1(1).EQ.IR(5).AND.IR1(2).EQ.IR(6))    GO TO 30
              IF(IR1(1).EQ.IR(7).AND.IR1(2).EQ.IR(8))    GO TO 40
              IF(IR1(1).EQ.IR(9).AND.IR1(2).EQ.IR(10))   GO TO 50
              IF(IR1(1).EQ.IR(11).AND.IR1(2).EQ.IR(12))  GO TO 60
              IF(IR1(1).EQ.IR(13).AND.IR1(2).EQ.IR(14))  GO TO 70
              IF(IR1(1).EQ.IR(15).AND.IR1(2).EQ.IR(16))  GO TO 9999
C*****
C     ERROR CAUTION FOR INVALID DATA CARD
C*****
      IFLAG=2
      WRITE(IOOUT,105)
  105 FORMAT(21X,35H*****CAUTION***** INVALID DATA CARD)
      GO TO 10
   20         TIME=R1(1)
              ISRC=IFIX(R1(2))
      GO TO 10
   30         ROBS(1)=R1(1)
              ROBS(2)=R1(2)
              ROBS(3)=R1(3)
      GO TO 10
   40         RTAR(1)=R1(1)
              RTAR(2)=R1(2)
              RTAR(3)=R1(3)
              TARDEG=R1(4)
      GO TO 10
   50         RSRC(1)=R1(1)
              RSRC(2)=R1(2)
              RSRC(3)=R1(3)
              TIMGUN=R1(4)
      GO TO 10
   60         FOV=R1(1)
              WAVE=R1(2)
              RADSEE=R1(3)
              RADLOC=R1(4)
              RCTSEE=R1(5)
              RCTLOC=R1(6)
      GO TO 10
   70         TARLEN=R1(1)
              TARWID=R1(2)
              TARHGT=R1(3)
              PCTSEE=R1(4)
              PCTLOC=R1(5)
   10 CONTINUE
      GO TO 9999
C*****
C*****CAUTION FOR TOO MANY CARDS
C*****
   90 WRITE(IOOUT,106)
      IFLAG=3
  106 FORMAT(21X,17H*****CAUTION*****,/,
     *21X,56HMORE THAN 10 DATA CARDS ENTERED--REMAINING CARDS IGNORED)
      GO TO 9999
  998 IFLAG=4
 9999 RETURN
      END
```

```
      SUBROUTINE DATWT(ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)
C*********************************************************************
C*                        SUBROUTINE DATWT                           *
C*                          FLASH MODULE                             *
C*                            EOSAEL80                               *
C*********************************************************************
      COMMON/IOUNT/IOIN,IOOUT
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
     *             TIME,ISRC,RCTSEE,RCTLOC
      DIMENSION I1(3),I2(3)
      DATA I1/2HNO,2HYE,2HPA/
      DATA I2/2H  ,2HS ,2HRT/
      IANS1=ISEE+1
      IF(ISEE.EQ.3)IANS1=3
      IF(ILOC.EQ.0)IANS2=2
      IF(ILOC.EQ.1)IANS2=1
      IF(ILOC.EQ.2)IANS2=3
      IF(ILOC.EQ.0)IANS2=2
      IF(ILOC.EQ.3)IANS2=3
   82 FORMAT(21X,36HDETECTION EXPOSURE CRITERIA DEFEATED)
   83 FORMAT(21X,38HDETECTION RESOLUTION CRITERIA DEFEATED)
   92 FORMAT(21X,34HLOCK ON EXPOSURE CRITERIA DEFEATED)
   93 FORMAT(21X,36HLOCK ON RESOLUTION CRITERIA DEFEATED)
   97 FORMAT(21X,17HTARGET OBSCURED ?,1X,2A2,/,
     *       21X,15HTARGET LOCKED ?,1X,2A2)
   98 FORMAT(21X,10HTIME LEFT:,4X,F8.3,1X,7HSECONDS,/,
     *       21X,13HTIME NO LOCK:,1X,F8.3,1X,7HSECONDS)
   99 FORMAT(21X,25HPROGRAM FLASH--STOP CODE:,1X,3I1)
  100 FORMAT(21X,11HSOURCE TYPE,1X,I2,1X,12HUNIDENTIFIED)
  101 FORMAT(21X,17HINPUT WAVELENGTH:,1X,F6.1,1X,11HMICROMETERS,
     *25H IS OUT OF RANGE OF MODEL)
  102 FORMAT(21X,37HGUNFLASH HAS NOT OCCURED YET: TIME = ,1X,F8.3,/,42X,
     *16HTIME OF FLASH = ,1X,F8.3)
  103 FORMAT(21X,38HFLASH IS NOT IN DETECTOR FIELD OF VIEW)
  104 FORMAT(21X,24HFLASH IS BEHIND TARGET--,
     *15H NO OBSCURATION)
  105 FORMAT(21X,24HNORMAL PROGRAM EXECUTION)
      WRITE(IOOUT,99)ISTOP,ISEE,ILOC
      IF(ISTOP.GT.1)GO TO 1
      IF(ISTOP.GT.0)GO TO 10
      WRITE(IOOUT,100)ISRC
      GO TO 9999
   10 WRITE(IOOUT,101)WAVE
      GO TO 9999
    1 WRITE(IOOUT,97)I1(IANS1),I2(IANS1),I1(IANS2),I2(IANS2)
      IF(ISEE.LT.2.AND.ILOC.LT.2)GO TO 11
      IF(ISEE.EQ.2)WRITE(IOOUT,82)
      IF(ISEE.EQ.3)WRITE(IOOUT,83)
      IF(ILOC.EQ.2)WRITE(IOOUT,92)
      IF(ILOC.EQ.3)WRITE(IOOUT,93)
      GO TO 9998
   11 IGO=ISTOP-1
      GO TO (2,3,4,5,6,7,8,9),IGO
    2 WRITE(IOOUT,102)TIME,TIMGUN
      GO TO 9999
    3 WRITE(IOOUT,103)
      GO TO 9999
    4 WRITE(IOOUT,103)
      GO TO 9999
    5 WRITE(IOOUT,104)
      GO TO 9999
    6 WRITE(IOOUT,103)
      GO TO 9999
    7 WRITE(IOOUT,103)
      GO TO 9999
    8 WRITE(IOOUT,103)
      GO TO 9999
    9 WRITE(IOOUT,105)
 9998 WRITE(IOOUT,98)TIMLEF,TIMNOL
```

```
9999 RETURN                                                    DAT00710
     END                                                       DAT00720
```

```
      SUBROUTINE GETIM(FLASHR,FLASHT,ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)
C***********************************************************************
C*                    SUBROUTINE GETIM                                 *
C*                         FLASH MODULE                                *
C*                         EOSAEL80                                    *
C***********************************************************************
      REAL LOCSTO,LOCFRA
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
     *             TIME,ISRC,RCTSEE,RCTLOC
      COMMON/CONST/PI
C*****ALL DISTANCES IN METERS ANGLES CONVERTED TO RADIANS(CCW A-AXES
      DIMENSION DOTVEC(3),RHTVEC(3),RHFVEC(3),RFLASH(3)
C*****SCALAR(A,B)=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)
      SCALAR(A1,A2,A3,B1,B2,B3)=A1*B1+A2*B2+A3*B3
C*****DEFAULT AND CONVERT INPUT
      ANGGUN=0.0
      ELEGUN=0.0
      GUNHT=0.0
      GUNLEN=0.0
      TARHHT=TARHGT/2.0
      ANGTAR=TARDEG*(PI/180.0)
      HAFFOV=(FOV/2.0)*(PI/180.0)
      SEESTO=RADSEE/1000.0
      LOCSTO=RADLOC/1000.0
      SEEFRA=PCTSEE/100.0
      LOCFRA=PCTLOC/100.0
      RADFL=FLASHR
      DURTIM=FLASHT
      DURSEE=10.0*RCTSEE
      DURLOC=10.0*RCTLOC
      REACFA=SEEFRA-LOCFRA
C*****INITIALIZE FLAGS *****
      AGINSE=0.0
      AGINLO=0.0
      TIMLEF=0.0
      TIMNOL=0.0
      TIMGON=TIME-TIMGUN
      ISEE=0
      ILOC=0
      ISTOP=0
      IF(ISRC.LT.1.OR.ISRC.GT.2)GO TO 9999
      ISTOP=1
      IWAVE=0
      IF(WAVE.GE.0.40.AND.WAVE.LE.0.70)IWAVE=1
      IF(WAVE.GE.8.00.AND.WAVE.LE.12.0)IWAVE=2
      IF(IWAVE.EQ.0)GO TO 9999
      ISTOP=2
      IF(TIMGON.LT.0.)GO TO 9999
      TIMLEF=DURTIM-TIMGON
C*****TIMLEF IS DURATION LEFT OF FLASH
      TIMNOL =TIMLEF+REACFA*DURTIM
C*****TIMNOL IS DURATION OF LOCK
      FLSHIF=RADFL
C*****CALCULATE FLASH COORDINATES FROM GIVEN COORDINATES
      RFLASH(1)=RSRC(1)+FLSHIF*COS(ANGGUN)*COS(ELEGUN)
      RFLASH(2)=RSRC(2)+FLSHIF*SIN(ANGGUN)*COS(ELEGUN)
      RFLASH(3)=RSRC(3)+FLSHIF*SIN(ELEGUN) + GUNHT
C*****DEFINE DIRECTION OF TARGET UNIT VECTOR
      DOTVEC(1)=COS(ANGTAR)
      DOTVEC(2)=SIN(ANGTAR)
      DOTVEC(3)=0.
C*****DEFINE OBSERVER-TARGET,OBSERVER-FLASH VECTORS
      DO 10 I=1,3
      RHTVEC(I)=RTAR(I)-ROBS(I)
      RHFVEC(I)=RFLASH(I)-ROBS(I)
   10 CONTINUE
C*****ADD HALF TARGET HEIGHT
      RHTVEC(3)=RHTVEC(3)+TARHHT
C*****FIND LENGTHS
```

```
      A1=RHTVEC(1)
      A2=RHTVEC(2)
      A3=RHTVEC(3)
      B1=RHFVEC(1)
      B2=RHFVEC(2)
      B3=RHFVEC(3)
      RHT=SQRT(SCALAR(A1,A2,A3,A1,A2,A3))
      RHF=SQRT(SCALAR(B1,B2,B3,B1,B2,B3))
C*****FIND COSINES BETWEEN VECTORS
      C1=DOTVEC(1)
      C2=DOTVEC(2)
      C3=DOTVEC(3)
      GAMMAD= SCALAR(C1,C2,C3,B1,B2,B3)/RHF
      GAMMAT= SCALAR(A1,A2,A3,B1,B2,B3)/(RHT*RHF)
      W     = SCALAR(C1,C2,C3,A1,A2,A3)/RHT
C*****FIND TRIG FCNS OF THETA=HALF-ANGLE OF FLASH CONE
      SINTH=RADFL/RHF
      COSTH=SQRT(1.0-SINTH*SINTH)
      COSSQ=COSTH*COSTH
      COSINE=ABS(W)
      SINE=SQRT(1.0-COSINE**2)
C*****CALCULATE CYLINDRICAL TARGET LONGEST DIMENSION AS SEEN IN
C*****PLANE PERPENDICULAR TO RHFVEC
      TARDIM=TARLEN*SINE+TARWID*COSINE
      HAFDIM=TARDIM/2.
      THETA=ATAN2(SQRT(1.0-COSTH**2),COSTH)
      THETAT=ATAN2(SQRT(1.0-GAMMAT**2),GAMMAT)
      DELTH=THETAT-THETA
      ISTOP=3
      IF(DELTH.GT.HAFFOV)GO TO 9999
C*****IF OUT OF FIELD OF VIEW,RETURN.
      DISPLC=RHT*SIN(DELTH)
      PROJRH=RHT*GAMMAT
      ISTOP=4
      IF(DISPLC.GT.HAFDIM)GO TO 99
      ISTOP=5
      IF(PROJRH.LT.RHF.AND.IWAVE.LT.2)GO TO 99
C*****NO OBSCURATION IF TARGET IN FRONT OF FLASH SO RETURN
C*****FROM HERE ON, GAMMAT NECESSARILY POSITIVE
C*****FIND INTERSECTIONS OF FLASH CONE WITH DOTVEC EXTENDED FROM TARGET
C*****GET COEFFICIENT OF QUADRATIC EQUATION FOR DISTANCE ALONG DOTVEC
C*****FROM TARGET TO FLASH CONE/RHT
      A  =COSSQ  -GAMMAD**2
      BD2=COSSQ*W-GAMMAD*GAMMAT
      C  =COSSQ  -GAMMAT**2
      B=BD2*2.
C*****IF DISCRIMINANT NEGATIVE, NO INTERSECTION
      DISCRM=BD2*BD2-A*C
      ISTOP=6
      IF(DISCRM.LT.0.)GO TO 99
C*****IF A=0,QUADRATIC FORMULA BLOWS UP.ACTUALLY HAVE LINEAR EQN.
      IF(ABS(A).GT.1.E-30)GOTO20
      IF(ABS(B).LT.1.E-30)B=SIGN(1.E-30,B)
      SPL=-C/B*RHT
      ZPL=SPL*GAMMAD+PROJRH
      ISTOP=7
      IF(ZPL.LE.0.) GO TO 99
C*****REJECT IF SOLE INTERSEC IS W NEG CONE SHEET.MEANINGLESS
      DPL=ABS(SPL)*SINE
      IF(GAMMAT-COSTH)14,14,12
C*****IF TAR OUTSIDE FLASH CONE GO TO 14,OTHERWISE 12
   12 OBSCUR=HAFDIM+AMIN1(DPL,HAFDIM)
      GO TO 30
   14 OBSCUR=AMAX1(0.,HAFDIM-DPL)
      GO TO 30
   20 ROOT=SQRT(DISCRM)
      ROOT=SIGN(ROOT,A)
C*****ABOVE NOT REALLY NEEDED BUT NICER TO HAVE SPL.GT.SMI-SEE BELOW
      SPL=(-BD2+ROOT)/A *RHT
      SMI=(-BD2-ROOT)/A *RHT
```

```
C*****SPL,SMI ARE 2 LENGTHS ALONG DOTVEC OF INTERSEC PTS W FLASH CONE     GTM01410
C*****ZPL,ZMI ARE Z COORDINATES OF INTERSECTIONS(ON FLASH CONE AXIS)      GTM01420
      ZPL=SPL*GAMMAD+PROJRH                                               GTM01430
      ZMI=SMI*GAMMAD+PROJRH                                               GTM01440
      ISTOP=8                                                             GTM01450
      IF(ZPL.LE.0..AND.ZMI.LE.0.)GO TO 99                                 GTM01460
C*****REJECT IF BOTH INTERCEPTS IN NEGATIVE CONE                          GTM01470
C*****DPL,DMI ARE PROJECNS OF SPL,SMI PERPEN TO RHTVEC                    GTM01480
C*****THEIR MAGNITUDES ARE LIMITED TO HAFDIM                              GTM01490
      DPL=SPL*SINE                                                        GTM01500
      DPL=AMIN1(DPL,HAFDIM)                                               GTM01510
      DPL=AMAX1(DPL,-HAFDIM)                                              GTM01520
      DMI=SMI*SINE                                                        GTM01530
      DMI=AMIN1(DMI,HAFDIM)                                               GTM01540
      DMI=AMAX1(DMI,-HAFDIM)                                              GTM01550
      OBSCUR=ABS(DPL-DMI)                                                 GTM01560
      IF(ZPL*ZMI.GT.0.)GOTO30                                             GTM01570
C*****SKIP AROUND  UNLESS BOTH SHEETS OF CONE INVOLVED                    GTM01580
      IF(ZPL.LE.0.)DPL=SIGN(HAFDIM,DPL)                                   GTM01590
      IF(ZMI.LE.0.)DMI=SIGN(HAFDIM,DMI)                                   GTM01600
      OBSCUR=TARDIM-ABS(DPL-DMI)                                          GTM01610
C*****ABOVE BRANCH RARE.INTERSEC W BOTH CONE SHEETS.NEG SHEET IGNORE.     GTM01620
   30 CONTINUE                                                            GTM01630
      SEEN=TARDIM-OBSCUR                                                  GTM01640
      ISTOP=9                                                             GTM01650
   98 IF(IWAVE.EQ.2)GO TO 1                                               GTM01660
      CALL VSBLC(ISEE,ILOC,SEEN,TARDIM,RHT,TIMLEF,SEESTO,LOCSTO,          GTM01670
     *           SEEFRA,LOCFRA)                                           GTM01680
      GO TO 99                                                            GTM01690
    1 CALL IRBLC(RHF,ISEE,ILOC,AGINSE,AGINLO,DURSEE,DURLOC,TIMGON)        GTM01700
   99 CONTINUE                                                            GTM01710
      TIMLEF =AMAX1(TIMLEF,AGINSE)                                        GTM01720
      TIMNOL=AMAX1(TIMNOL,AGINLO)                                         GTM01730
 9999 RETURN                                                              GTM01740
      END                                                                 GTM01750
```

```
      SUBROUTINE IRBLC(RHF,ISEE,ILOC,AGINSE,AGINLO,DURSEE,DURLOC,TIMGON)IRBLC010
C*********************************************************************IRBLC020
C*                    SUBROUTINE IRBLC                               *IRBLC030
C*                         FLASH MODULE                              *IRBLC040
C*                         EOSAEL80                                  *IRBLC050
C*********************************************************************IRBLC060
      INTEGER LOCIRM                                                    IRBLC070
      SEEIRM=4.0                                                        IRBLC080
      LOCIRM=5.0                                                        IRBLC090
      SQRTRH=SQRT(RHF)                                                  IRBLC100
      AGINSE=DURSEE/SQRTRH                                              IRBLC110
      AGINLO=DURLOC/SQRTRH                                              IRBLC120
C*****ABOVE WHOLLY EMPIRICAL FROM FIT TO TV TAPES AT 2 RANGES           IRBLC130
      IF(AGINSE.GT.SEEIRM) AGINSE=SEEIRM                                IRBLC140
      IF(AGINLO.GT.LOCIRM) AGINLO=LOCIRM                                IRBLC150
      IF(AGINSE.GT.TIMGON) ISEE=1                                       IRBLC160
      IF(AGINLO.GT.TIMGON) ILOC=1                                       IRBLC170
      RETURN                                                            IRBLC180
      END                                                               IRBLC190
```

```
      SUBROUTINE VSBLC(ISEE,ILOC,SEEN,TARDIM,RHT,TIMLEF,SEESTO,LOCSTO,
     *                 SEEFRA,LOCFRA)
C***********************************************************************
C*                    SUBROUTINE VSBLC                                 *
C*                       FLASH MODULE                                  *
C*                       EOSAEL80                                      *
C***********************************************************************
      REAL LOCCRI,LOCFRA,LOCSTO
C*****SEEFRA IS FRACTION THAT MUST BE SEEN,LIKEWISE FOR LOCK
C*****SEESTO IS MIN ANGLE THAT MUST BE SEEN,LIKEWISE FOR LOCK
      SEEANG=SEEN/RHT
      TARANG=TARDIM/RHT
C*****DEFINE SEEN ANGLE AND TARGET ANGLE
      SEECRI=AMIN1(TARANG,SEESTO)
      LOCCRI=AMIN1(TARANG,LOCSTO)
      IF(TIMLEF.LE.0.)GO TO 9999
      IF(SEEANG.LT.SEEFRA*TARANG)GO TO 48
      ISEE=2
      GO TO 50
   48 IF(SEEANG.LT.SEECRI)GO TO 49
      ISEE=1
      GO TO 50
   49 CONTINUE
      ISEE=3
   50 CONTINUE
      IF(SEEANG.LT.LOCFRA*TARANG)GO TO 98
      ILOC=2
      GO TO 9999
   98 IF(SEEANG.LT.LOCCRI)GO TO 99
      ILOC=1
      GO TO 9999
   99 CONTINUE
      ILOC=3
 9999 RETURN
      END
```

```
EORUN    ,     4.0
WAVL     ,    10.6     10.6
VIS      ,     7.0
TURB
XSCALE   ,     3.0
SMOKE
DRTRAN
LZTRAN
CLTRAN   ,     3.0
SCREEN
OVRCST   ,     2.0
CLIMAT   ,     1.0      1.0     5.0      6.0    -1.0     1.0
GRNADE
GO
PARM          0.1016    1.3E-04         0.0      400.0        5.0      512.0
CN1              1.0    6.0E-13   1.0E-14    1.3E-14    2.7E-14    5.0E-14
V1               1.0       0.93      0.93       0.93       0.93       0.93
DVRV          0.400       500.0
CN2              1.0    6.0E-13   1.0E-14    1.3E-14    2.7E-14    5.0E-14
V2               1.0       0.93      0.93       0.93       0.93       0.93
GO
END
FOG        1.
HORZ     0.4
GO
FOG        2.
SLNH       0.133,56.3
GO
FOG        3.
HORZ     0.4
GO
END
MUNC           0.0       -50.0         0.
OBSC       200.         0.            2.0
TARC      -200.         0.            2.0
BART       5.           180.          5.          90.
OUTP       0.           0.            0.           0.
BURN       10.          0.            0.          0.          0.
GO
BURN        4.          0.            0.          0.          0.
BART       5.           250.          5.          90.
GO
DONE
END
MET1           4.       2.     286.0        2.      2.0      0.0
MET2           1.       53.
SOIL           2.       0.0     0.15
CHAR           3.       6.8     0.0
EXPL           1.       1.       1.      0.0      0.0      0.0      0.0      0.0      0.0
OBSC       200.0        0.0     2.0
TRNC      -200.0        0.0     2.0
RECC       200.0        0.0     2.0
TIMS         1.0       71.0     2.0
GO           1.
DONE
END
             8.55        15.        0.4
END
SEEK,       0.2,         0.0,         0.6
TARG,      -0.2,         0.0,       0.002
CLST,       0.20,       0.40
GO
SEEK,     0.133,         0.0,         0.5
GO
SEEK,       0.0,         0.0,         0.3
GO
END
 1 1 1
TARV            1.0        2.0       0.24       2.3       81.       2.      .000
SENS            .99        8.        1.         0.        0.        1.
```

```
GO
SENS              .90        8.        1.        0.        0.        1.
GO
SENS              .75        8.        1.        0.        0.        1.
GO
SENS              .50        8.        1.        0.        0.        1.
GO
SENS              .10        8.        1.        0.        0.        1.
GO
DONE
SCRN          15.     400.      0.4        0.        90.       74.       2.
DONE
END
OPOS,    -0.0667,       0.0,       0.2
CLDS,        0.2,     200.0,      40.0,       0.7,       1.0
SPOS,       -0.2,       0.0,     0.002
BKGR,       50.0
GRND,       50.0
TEMP,        9.8
GO
OPOS,    -0.1333,       0.0,       0.1
GO
END
NAME
OUTP
OBSC     200.0     0.0       2.0
MUNC    -200.0       0.0     2.0       95.0      100.0      10.0
TARC    -200.0     +40.      2.0
BART     5.0       400.0     5.0       90.0
MUNT     1.0       0.793     14.3      1.0       0.0        4.7       0.07
METR     50.0      2.0       220.0     4.0       20.0       0.0       0.0
EXTC
BURN
MISC
GO
DONE
END
WAVL    ,   1.06    1.06
VIS     ,    5.0     5.0    4.0
BASCAT
FCLOUD
GO
PART,       1.,     5000.,        1.
SORC,     -0.2,        0.,   -0.098,       90.,        0.,       50.
DETR,      0.2,        0.,   -0.098,       90.,      180.,        1.,      1.
CLDS,       .1,        .2,       .1,        0.,        0.,        0.,      5.
GRND,     -0.1,       0.5
PULS,      .33,        0.
GO
END
CPOS,      0.0,       0.0,       0.1
RPOS,      0.2,       0.0,     0.002
SPOS,     -0.2,       0.0,     0.002
AXES,      0.1,       0.2,       0.1
CLDS,      5.0,      0.95,       4.0,       1.0,       9.8
ATMO,      2.0,       9.8
BKGR,      0.5,      50.0
SANG,     80.0,       0.0
LUND,      0.0
GO
END
FREQ    ,   35.0    35.0
NMMW    ,    2.0
GO
PATH,      0.4
ATMO,     15.0    1013.25      6.44
FOGD,      0.5
RAIN,      5.0
GO
ATMO,     -1.0    1015.2       7.7
```

```
RAIN,        0.0
SNOW,        5.0
GO
END
WVNUM        2010.  2710.  2.
LT4M
RESF
SPOT
GO
ENVR          3.         2.         2.         4.         1.         1.
EMIS          .100+01    .283+03    .950+00    .295+03
ATM           .650+02    0.0        .0
TARG          .400+00    .450+02    .900+02    .450+02
REFL         0.5        0.5         .0         .500-01    .0         0.
SENS          .200-02    .900+02    .270+03    .100+01
GO
12
        3.5,        0.78
        3.6,        0.83
        3.7,        0.87
        3.8,        0.92
        3.9,        0.96
        4.0,        0.98
        4.1,        0.97
        4.3,        0.96
        4.5,        0.95
        4.7,        0.94
        4.9,        0.93
        5.0,        0.93
END
   4   2   1   0   1   0   0   0   0      0.000      0.000
       0.002      0.002      0.000      0.400      0.000      1.000
   4
   4   2   1   0   1   0   0   0   1      0.000      0.000
   0
END
```

```
******************************
*   ELECTRO-OPTICAL SYSTEMS  *
*   ATMOSPHERIC EFFECTS      *
*        LIBRARY             *
******************************

INDIVIDUAL MODULES  SELECTED
           TURB
           XSCALE
           SMOKE
           DRTRAN
           LZTRAN
           CLTRAN
           SCREEN
           OVRCST
           GRNADE
           CLIMATE

                         BEGINNING          ENDING
WAVENUMBER(CM**-1)         943.396         943.396
WAVELENGTH(MICRONS)         10.600          10.600
FREQUENCY(GHZ)           28301.886       28301.886
```

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

```
CLASS  1 = FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS  2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLASS  3 = FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLASS  4 = FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS  5 = DUST WITH VIS LT 3 KM.
CLASS  6 = DUST WITH VIS GE 3 KM.
CLASS  7 = DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM.
CLASS  8 = DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM.
CLASS  9 = DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM.
CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM.
CLASS 11 = SNOW WITH VIS LT 1 KM.
CLASS 12 = SNOW WITH 1 LE VIS LT 3 KM.
CLASS 13 = SNOW WITH 3 LE VIS LT 7 KM.
CLASS 14 = SNOW WITH VIS GE 7 KM.
CLASS 15 = NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 16 = NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 17 = VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 18 = VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLASS 19 = CEILING HEIGHT LT 300 M.
CLASS 20 = CEILING HEIGHT LT 1000 M.
CLASS 21 = NO CEILING.
CLASS 22 = ALL CONDITIONS COMBINED.
```

EOSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS DURING SPRING AT 03-09 (LST).

CLASS NO.	FREQCY CLASS (%)	MEAN TEMP (C)	MEAN DP (C)	MEAN AH (GM/CU.M)	MEAN RH (%)	MEAN VIS (KM)	MEAN PRESS (MB)	MEAN/STDEV WNDVEL (MPS)	MEAN CLDHT (KM)	MEAN/STDEV CLDCVR (%)	FREQCY A (%)	FREQCY B (%)	FREQCY C (%)	FREQCY D (%)	FREQCY E (%)	FREQCY F (%)
1	4.5	3.5	3.2	6.3	98.2	.365	1016.7	1.9/ 1.7	.929	74.4/ 39.2	.0	2.6	12.9	69.2	1.4	14.0
2	8.2	4.1	3.2	6.4	93.8	1.945	1016.1	2.3/ 1.7	1.818	60.0/ 40.4	.0	4.1	19.0	52.6	5.1	19.2
3	17.9	5.1	3.3	6.4	89.0	4.502	1016.4	2.8/ 2.1	2.124	57.8/ 39.4	.0	4.3	15.4	52.4	8.9	19.0
4	9.6	5.8	3.3	6.3	84.5	9.224	1016.0	3.4/ 2.2	1.911	56.2/ 38.6	.0	3.1	12.5	53.1	11.5	19.7
5	.0	.0	.0	.0	.0	.000	.0	.0/ .0	.000	.0/ .0	.0	.0	.0	.0	.0	.0
6	.0	.8	-1.4	5.5	85.5	10.617	1012.3	4.2/ 3.7	1.279	64.0/ 38.2	.0	.0	5.9	64.7	.0	29.4
7	.6	5.9	5.5	7.3	97.6	.460	1013.4	2.9/ 2.6	.204	98.5/ 10.3	.0	.0	.3	97.8	.3	1.7
8	2.7	6.6	5.8	7.5	95.2	1.955	1011.8	3.6/ 2.3	.565	98.4/ 8.7	.0	.1	1.9	95.4	1.6	1.0
9	5.8	6.8	5.6	7.3	92.4	4.590	1010.3	4.5/ 2.7	.872	98.0/ 7.3	.0	.2	2.2	94.5	1.7	1.5
10	7.1	6.9	4.9	7.0	88.0	12.633	1008.8	5.6/ 3.0	1.340	92.7/ 14.5	.0	.1	1.8	92.8	3.5	1.8
11	.2	-1.0	-1.6	4.4	95.9	.522	1009.9	4.3/ 3.2	.428	96.9/ 13.5	.0	.9	2.7	95.5	.0	.9
12	.7	-1.1	-1.8	4.4	95.0	1.870	1009.9	4.4/ 2.7	.270	97.4/ 10.7	.0	.2	2.2	96.7	.2	.7
13	.8	-1.4	-2.7	4.1	91.2	4.436	1013.4	4.6/ 2.7	.824	94.1/ 14.4	.0	.0	2.7	93.0	2.3	[illegible]
14	.8	-.5	-2.5	4.2	86.9	13.761	1010.9	5.6/ 3.7	1.359	83.8/ 22.9	.0	.0	1.7	91.1	4.5	[illegible]
15	40.3	5.2	2.3	5.9	82.3	14.756	1015.9	3.9/ 2.6	2.293	51.9/ 37.9	.0	2.1	10.1	54.8	13.4	[illegible]
16	.8	14.7	12.5	11.1	87.3	12.319	1011.7	3.2/ 1.9	2.938	66.5/ 29.8	.0	6.9	17.7	47.0	[illegible]	[illegible]
17	3.6	4.0	3.8	6.6	98.4	.343	1015.5	2.4/ 1.9	.088	99.0/ 4.9	.0	.0	1.8	97.1	.4	[illegible]
18	10.0	4.6	4.0	6.7	96.4	1.312	1014.2	3.1/ 2.2	.189	98.0/ 6.3	.0	.0	2.6	94.9	1.2	1.4
19	14.2	4.9	4.2	6.7	95.3	3.326	1013.7	3.9/ 2.6	.143	98.1/ 6.0	.0	.0	1.2	97.0	1.0	[illegible]
20	35.4	5.2	3.8	6.5	90.9	7.312	1012.9	4.4/ 2.8	.390	94.9/ 9.4	.0	.0	2.0	93.8	2.7	[illegible]
21	37.2	4.2	1.8	5.8	85.1	10.680	1018.1	2.9/ 2.3	1.878	16.4/ 17.2	.0	5.6	17.9	28.5	13.4	[illegible]
22	100.0	5.3	3.2	6.3	87.1	9.302	1015.0	3.6/ 2.6	1.885	63.2/ 38.7	.0	2.5	10.7	61.9	9.1	[illegible]

CLASS NO.	FREQCY CLASS (%)	FREQCY WNDDIR 015 (%)	FREQCY WNDDIR 045 (%)	FREQCY WNDDIR 075 (%)	FREQCY WNDDIR 105 (%)	FREQCY WNDDIR 135 (%)	FREQCY WNDDIR 165 (%)	FREQCY WNDDIR 195 (%)	FREQCY WNDDIR 225 (%)	FREQCY WNDDIR 255 (%)	FREQCY WNDDIR 285 (%)	FREQCY WNDDIR 315 (%)	FREQCY WNDDIR 345 (%)	FREQCY WNDDIR VBL (%)
1	4.5	8.6	11.1	8.3	4.0	2.8	4.6	5.2	6.9	8.2	7.0	4.7	7.2	21.3
2	8.2	8.5	12.0	10.0	5.9	4.6	5.7	5.7	8.7	8.5	5.8	3.9	5.9	14.8
3	17.9	7.6	10.1	11.4	8.3	6.7	7.0	7.7	10.1	8.4	5.3	3.6	5.3	8.5
4	9.6	6.3	7.7	9.8	7.8	7.4	8.6	9.5	11.3	11.[illegible]	6.8	3.9	4.7	5.1
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	23.5	11.8	.0	5.9	5.9	11.8	11.8	5.9	5.9	5.9	11.8
7	.6	8.9	11.1	10.0	3.9	1.4	2.5	4.4	12.2	12.2	10.6	7.2	9.4	6.1
8	2.7	6.5	7.8	6.9	4.2	3.3	4.9	8.3	15.6	14.5	9.7	6.8	5.3	6.0
9	5.8	5.3	6.6	4.7	3.6	4.0	6.8	11.3	19.5	14.9	9.5	6.1	5.1	2.5
10	7.1	4.2	4.1	2.6	2.3	3.6	7.3	12.3	21.7	16.9	11.5	7.4	5.0	1.1
11	.2	10.0	12.7	3.6	1.8	1.8	1.8	3.6	3.6	13.6	12.7	11.8	10.9	11.8
12	.7	11.1	12.9	7.3	3.8	3.1	2.9	4.5	8.2	13.8	10.7	7.8	12.0	1.8
13	.8	10.1	11.5	9.5	3.5	1.9	2.7	4.1	9.3	15.4	10.1	7.8	10.7	3.5
14	.8	7.0	6.8	7.4	3.2	.6	1.9	1.9	8.5	21.9	17.5	12.1	8.7	2.6
15	40.3	7.5	9.1	9.7	6.8	4.7	6.8	8.4	13.2	10.5	7.8	5.0	5.9	4.8
16	.8	3.0	4.5	4.3	10.2	10.8	20.7	11.4	14.0	7.3	2.8	2.8	1.8	6.3
17	3.6	10.0	12.9	8.2	3.8	2.5	3.3	5.0	8.8	10.0	8.8	6.1	8.6	12.1
18	10.0	9.3	12.0	8.0	3.6	2.5	3.5	5.4	11.2	12.6	9.6	6.4	7.6	8.3
19	14.2	9.2	10.9	7.5	2.9	1.7	2.7	5.4	12.7	14.8	10.7	7.9	8.3	5.3
20	35.4	7.5	8.2	6.2	2.9	2.1	4.0	7.6	16.1	16.1	11.6	7.3	7.2	3.2
21	37.2	8.1	10.8	12.7	8.7	6.3	7.2	7.1	8.4	6.9	5.0	3.6	5.2	9.9
22	100.0	7.2	8.9	9.0	6.3	5.0	6.8	8.3	12.7	10.9	7.6	4.9	5.7	6.6

VISIBILITY
9.30 KM

TUPB LASER MODULE

CALCULATION OF POWER SPECTRUM AND TURBULENCE INDUCED POINTING JITTER OF A LASER TARGET DESIGNATOR AND SEEKER

LASER WAVELENGTH (MICROMETERS)	10.6000
DESIG. APERTURE DIAMETER (METERS)	.101600
BEAMSPREAD ANGLE (RADIANS)	.000130
SEEKER APERTURE DIAMETER (METERS)	.400000
RANGE FROM TARGET TO SEEKER (METERS)	500.00
BEAM SLUE RATE (RAD/SEC)	.000000
DESIGNATION RANGE (METERS)	400.00
DURATION OF TEST (SECONDS)	5.0000
TOTAL DESIGNATOR PATH SEGMENTS	5
TOTAL SEEKER PATH SEGMENTS	5
TOTAL FREQUENCIES FOR WHICH POWER SPECTRUM IS TO BE CALCULATED	512

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN DESIGNATOR PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-013	.93
3	.130000-013	.93
4	.270000-013	.93
5	.500000-013	.93

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN SEEKER PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-013	.93
3	.130000-013	.93
4	.270000-013	.93
5	.500000-013	.93

DESIGNATOR TO TARGET

VIRTUAL POINT SOURCE TO APERTURE DISTANCE		781.53845 (METERS)
DISTANCE FROM VIRTUAL POINT SOURCE TO TARGET		1181.53845 (METERS)
INTEGRATED COHERENCE LENGTH		.335957 (METERS)
DIAMETER/INTEGRATED COHERENCE LENGTH		.302419
TRANSMITTER-INDUCED BEAM SPREAD	.13000-003 (RADIANS)	.052000 (METERS)
DIFFRACTION-LIMITED BEAM SPREAD	.11769-003 (RADIANS)	.047074 (METERS)
DIFFRACTION AND TURBULENCE BEAM SPREAD	.11907-003 (RADIANS)	.047629 (METERS)
TOTAL EFFECTIVE BEAM SIZE	.17629-003 (RADIANS)	.172116 (METERS)

SEGMENT NO.	COHERENCE LENGTH	REFERENCE FREQUENCY(HERTZ)
1	.341660	56.928500
2	5.124240	18.976166
3	6.129030	11.385700
4	6.588533	8.132643
5	13.656743	6.325389

TARGET TO SEEKER

INTEGRATED COHERENCE LENGTH		.934983 (METERS)
DIAMETER/INTEGRATED COHERENCE LENGTH		.184085
TRANSMITTER-INDUCED BEAM SPREAD	.13000-003 (RADIANS)	.065000 (METERS)
DIFFRACTION-LIMITED BEAM SPREAD	.69469-004 (RADIANS)	.034735 (METERS)
DIFFRACTION AND TURBULENCE BEAM SPREAD	.69828-004 (RADIANS)	.034914 (METERS)
TOTAL EFFECTIVE BEAM SIZE	.14757-003 (RADIANS)	.245900 (METERS)

SEGMENT NO.	COHERENCE LENGTH	REFERENCE FREQUENCY(HERTZ)
1	2.689629	7.400705
2	10.458297	2.466902
3	5.360999	1.480141
4	2.469819	1.057244
5	1.327262	.822301

THE VARIANCE OF THE POWER SPECTRUM IS .8090-011

OUTPUT FOR DESIGNATOR TO TARGET PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .92021-007, MEAN OF IMAG PART = .00000
VAR. OF REAL PART = .83284-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = -.52602-011, MEAN OF IMAG PART = .19630-016
VAR. OF REAL PART = .79425-017, VAR. OF IMAG PART = .13959-031

OUTPUT FOR TARGET TO SEEKER PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .17265-006, MEAN OF IMAG PART = .00000
VAR. OF REAL PART = .91505-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = .24531-011, MEAN OF IMAG PART = .26733-016
VAR. OF REAL PART = .87592-017, VAR. OF IMAG PART = .12492-031

```
XSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE


                              OPTIONS CHOSEN
                              MARITIME ARTIC
                              HORIZONTAL PATH
             EXTINCTION FROM 8.0 TO 12.0 MICRONS   DISTANCE   TRANSMISSION
                           KM**-1                     KM
                             .178                    .400       .931+000

                              OPTIONS CHOSEN
                              MARITIME POLAR
                              SLANT PATH FOR    10.600 MICRONS
WARNING FROM SLANT
THE VERTICAL DISTANCE EXCEEDS THE 500 METER UPPER LIMIT, OR
IS NOT AN INTEGER MULTIPLE OF 20 METERS
SLANT DISTANCE CHANGED FROM   .2397 TO   .2238 KM
           SLANT EXTINCTION AT 10.60 MICRONS   DISTANCE   TRANSMISSION      ANGLE
                           KM**-1                 KM
                            1.295                .224       .748+000        56.30

                              OPTIONS CHOSEN
                              CONTINENTAL POLAR
                              HORIZONTAL PATH
             EXTINCTION FROM 8.0 TO 12.0 MICRONS   DISTANCE   TRANSMISSION
                           KM**-1                     KM
                             .008                    .400       .997+000
```

SMOKE MODEL MODULE

```
*****************
*               *
*     SMOKE     *          EXECUTION  1
*               *
*****************
```

SMOKE MUNITIONS
WHITE PHOSPHORUS (WP)
NO. ROUNDS 1.
FILL WEIGHT 15.600 LB
BURN TIME 1.0 SEC
EFFICIENCY 100.0 PERCENT
YIELD FACTOR 5.84

METEOROLOGICAL CONDITIONS

WINDSPEED	3.6	M/S
WIND DIRECTION (USUAL MET CONVENTION AZIMUTH)	225.0	DEGREES
RELATIVE HUMIDITY	87.1	PERCENT
PASQUILL CATEGORY	D	
AIR TEMPERATURE	5.3	DEGREE C
TEMP. GRADIENT	.00	C DEG./M

EXTINCTION COEFFICIENTS

0.4-0.7	MICROMETERS	4.304	M**2/GM
0.7-1.2	MICROMETERS	2.166	M**2/GM
1.06	MICROMETERS	1.541	M**2/GM
3.0-5.0	MICROMETERS	.350	M**2/GM
8.0-12.	MICROMETERS	.338	M**2/GM
10.6	MICROMETERS	.364	M**2/GM
94.0	GHZ	.001	M**2/GM

BURN RATE PROFILE = 1.0000

	FIELD COORDINATES (X)	(Y)	(Z)		ROTATED COORD.(WIND X-AXIS,MUNITION ORIGIN) (XW)	(YW)	(ZW)	
MUNITION COORDINATES=	.00	-50.00	.00	METERS	.00	.00	.00	METERS
OBSERVER COORDINATES=	200.00	.00	2.00	METERS	176.78	-106.07	2.00	METERS
TARGET COORDINATES=	-200.00	.00	2.00	METERS	-106.07	176.78	2.00	METERS

ANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = 90.00 DEG.
EVENT TIME = .0 SEC

TIME (SEC)	LENGTH (METERS)	WIDTH (METERS)	HEIGHT (METERS)	PATHLENGTH (METERS)	CL (GM/M**2)	TRANSMISSION SPECTRAL BANDS (MICROMETERS) 0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94.GHZ
5.	18.	13.	22.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10.	36.	16.	34.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15.	54.	20.	44.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20.	72.	23.	53.	18.16	4.01	.000	.000	.002	.245	.258	.232	.996
25.	90.	27.	61.	37.02	4.89	.000	.000	.001	.181	.191	.169	.995
30.	108.	31.	69.	40.38	3.48	.000	.001	.005	.296	.309	.282	.997
35.	126.	34.	76.	43.27	2.58	.000	.004	.019	.405	.418	.391	.997
40.	144.	38.	83.	45.86	1.99	.000	.013	.047	.498	.510	.485	.998
45.	162.	41.	90.	48.25	1.58	.001	.033	.088	.575	.586	.563	.998
50.	180.	45.	95.	50.49	1.29	.004	.062	.138	.638	.648	.626	.999
55.	198.	49.	101.	52.62	1.07	.010	.099	.193	.688	.697	.678	.999
60.	216.	52.	106.	54.64	.90	.021	.142	.250	.730	.738	.721	.999
65.	234.	56.	111.	56.58	.77	.036	.189	.305	.764	.771	.756	.999
70.	252.	59.	115.	58.45	.67	.057	.236	.358	.792	.798	.785	.999
75.	270.	63.	120.	60.26	.58	.081	.283	.407	.815	.821	.809	.999
80.	288.	67.	124.	62.02	.52	.109	.328	.452	.835	.840	.829	.999
85.	306.	70.	127.	63.72	.46	.139	.370	.493	.852	.856	.846	1.000
90.	324.	74.	131.	65.38	.41	.171	.411	.531	.866	.870	.861	1.000
95.	342.	77.	135.	67.00	.37	.203	.448	.565	.878	.882	.874	1.000
100.	360.	81.	138.	68.59	.34	.235	.483	.596	.889	.893	.885	1.000
105.	378.	85.	141.	70.13	.31	.267	.515	.624	.898	.902	.894	1.000
110.	396.	88.	144.	71.65	.28	.299	.545	.649	.906	.910	.903	1.000
115.	414.	92.	147.	73.14	.26	.330	.572	.672	.914	.917	.910	1.000
120.	432.	95.	150.	74.60	.24	.359	.597	.693	.920	.923	.917	1.000
125.	450.	99.	153.	76.04	.22	.387	.620	.712	.926	.928	.923	1.000
130.	468.	103.	155.	77.45	.20	.414	.642	.729	.931	.933	.928	1.000
135.	486.	106.	158.	78.83	.19	.440	.662	.745	.935	.938	.933	1.000
140.	504.	110.	160.	[illegible].20	.18	.465	.680	.760	.940	.942	.937	1.000
145.	522.	113.	163.	81.55	.17	.488	.697	.773	.943	.945	.941	1.000
150.	540.	117.	165.	82.87	.16	.510	.712	.786	.947	.948	.945	1.000
155.	558.	121.	167.	84.18	.15	.531	.727	.797	.950	.951	.949	1.000
160.	576.	124.	170.	85.47	.14	.550	.740	.808	.953	.954	.951	1.000
165.	594.	128.	172.	86.75	.13	.569	.753	.817	.955	.957	.953	1.000
170.	612.	131.	174.	88.00	.12	.587	.765	.826	.958	.959	.956	1.000
175.	630.	135.	176.	89.25	.12	.603	.776	.835	.960	.961	.958	1.000
180.	648.	139.	178.	90.48	.11	.619	.786	.842	.962	.963	.960	1.000

SMOKE

EXECUTION 2

SMOKE MUNITIONS
HC SMOKE
NO. ROUNDS 1.
FILL WEIGHT 17.190 LB
BURN TIME 100.0 SEC
EFFICIENCY 40.0 PERCENT
YIELD FACTOR 2.39

METEOROLOGICAL CONDITIONS
WINDSPEED 3.6 M/S
WIND DIRECTION (USUAL
MET CONVENTION AZIMUTH) 225.0 DEGREES
RELATIVE HUMIDITY 87.1 PERCENT
PASQUILL CATEGORY D
AIR TEMPERATURE 5.3 DEGREE C
TEMP. GRADIENT .00 C DEG./M

EXTINCTION COEFFICIENTS
0.4-0.7 MICROMETERS 4.573 M**2/GM
0.7-1.2 MICROMETERS 2.180 M**2/GM
1.06 MICROMETERS 2.040 M**2/GM
3.0-5.0 MICROMETERS .190 M**2/GM
8.0-12. MICROMETERS .052 M**2/GM
10.6 MICROMETERS .051 M**2/GM
94.0 GHZ .001 M**2/GM

BURN RATE PROFILE = .5370+ .4760 (T/TBURN) + 4.7790 (T/TBURN)**2 + -5.4720 (T/TBURN)**3

	FIELD COORDINATES (X)	(Y)	(Z)		ROTATED COORD.(WIND X-AXIS,MUNITION ORIGIN) (XW)	(YW)	(ZW)	
MUNITION COORDINATES=	.00	-50.00	.00	METERS	.00	.00	.00	METERS
OBSERVER COORDINATES=	200.00	.00	2.00	METERS	176.78	-106.07	2.00	METERS
TARGET COORDINATES=	-200.00	.00	2.00	METERS	-106.07	176.78	2.00	METERS

ANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = 90.00 DEG.
EVENT TIME = .0 SEC

TIME (SEC)	LENGTH (METERS)	WIDTH (METERS)	HEIGHT (METERS)	PATHLENGTH (METERS)	CL (GM/M**2)	TRANSMISSION SPECTRAL BANDS (MICROMETERS) 0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94.GHZ
5.	18.	13.	6.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10.	36.	16.	8.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15.	54.	20.	10.	.00	.01	.949	.975	.977	.998	.999	.999	1.000
20.	72.	23.	11.	15.56	1.16	.005	.079	.094	.802	.941	.942	.999
25.	90.	27.	12.	30.12	2.00	.000	.013	.017	.684	.901	.903	.998
30.	108.	31.	13.	28.24	2.31	.000	.006	.009	.645	.887	.889	.998
35.	126.	34.	14.	26.88	2.63	.000	.003	.005	.607	.872	.874	.997
40.	144.	38.	16.	25.86	2.98	.000	.001	.002	.567	.856	.859	.997
45.	162.	41.	17.	25.07	3.35	.000	.001	.001	.529	.840	.843	.997
50.	180.	45.	18.	24.42	3.73	.000	.000	.000	.492	.823	.827	.996
55.	198.	49.	19.	23.89	4.11	.000	.000	.000	.458	.807	.811	.995
60.	216.	52.	20.	23.45	4.47	.000	.000	.000	.427	.792	.796	.996
65.	234.	56.	21.	23.07	4.80	.000	.000	.000	.401	.779	.783	.995
70.	252.	59.	23.	22.75	5.09	.000	.000	.000	.380	.768	.772	.995
75.	270.	63.	24.	22.46	5.30	.000	.000	.000	.365	.759	.763	.995
80.	288.	67.	25.	22.22	5.44	.000	.000	.000	.356	.754	.758	.995
85.	306.	70.	26.	21.99	5.49	.000	.000	.000	.353	.752	.756	.995
90.	324.	74.	27.	21.80	5.42	.000	.000	.000	.357	.755	.759	.995
95.	342.	77.	29.	21.62	5.22	.000	.000	.000	.371	.762	.766	.995
100.	360.	81.	30.	21.46	4.87	.000	.000	.000	.396	.776	.780	.995
105.	378.	85.	31.	21.32	4.37	.000	.000	.000	.436	.797	.800	.996
110.	396.	88.	32.	21.18	3.68	.000	.000	.001	.497	.826	.829	.996
115.	414.	92.	33.	21.06	2.79	.000	.002	.003	.588	.865	.867	.996
120.	432.	95.	35.	20.95	1.28	.003	.060	.073	.783	.935	.937	.999
125.	450.	99.	36.	20.85	.62	.058	.257	.281	.889	.968	.969	.999
130.	468.	103.	37.	20.75	.47	.116	.358	.383	.915	.976	.976	.999
135.	486.	106.	38.	20.67	.37	.182	.443	.468	.932	.981	.981	1.000
140.	504.	110.	39.	20.58	.30	.251	.517	.540	.944	.984	.985	1.000
145.	522.	113.	40.	20.51	.25	.319	.579	.601	.954	.987	.987	1.000
150.	540.	117.	42.	20.44	.21	.384	.633	.652	.961	.989	.989	1.000
155.	558.	121.	43.	20.37	.18	.443	.678	.696	.967	.991	.991	1.000
160.	576.	124.	44.	20.30	.15	.498	.717	.733	.971	.992	.992	1.000
165.	594.	128.	45.	20.25	.13	.547	.750	.764	.975	.993	.993	1.000
170.	612.	131.	46.	20.19	.11	.591	.778	.791	.978	.994	.994	1.000
175.	630.	135.	48.	20.14	.10	.630	.802	.814	.981	.995	.995	1.000
180.	648.	139.	49.	20.09	.09	.665	.823	.834	.983	.995	.995	1.000
185.	666.	142.	50.	20.04	.08	.696	.841	.851	.985	.996	.996	1.000
190.	684.	146.	51.	19.99	.07	.723	.857	.866	.987	.996	.996	1.000
195.	702.	149.	52.	19.95	.06	.748	.870	.879	.988	.997	.997	1.000
200.	720.	153.	54.	19.91	.06	.770	.882	.890	.989	.997	.997	1.000
205.	738.	157.	55.	19.87	.05	.789	.893	.900	.990	.997	.997	1.000
210.	756.	160.	56.	19.83	.05	.806	.902	.909	.991	.998	.998	1.000
215.	774.	164.	57.	19.80	.04	.822	.911	.916	.992	.998	.998	1.000
220.	792.	167.	58.	19.77	.04	.836	.918	.923	.993	.998	.998	1.000
225.	810.	171.	59.	19.73	.04	.849	.925	.930	.993	.998	.998	1.000
230.	828.	175.	61.	19.70	.03	.860	.931	.935	.994	.998	.998	1.000
235.	846.	178.	62.	19.67	.03	.871	.936	.940	.994	.998	.998	1.000
240.	864.	182.	63.	19.64	.03	.880	.941	.945	.995	.999	.999	1.000
245.	882.	185.	64.	19.62	.03	.888	.945	.949	.995	.999	.999	1.000
250.	900.	189.	65.	19.59	.02	.896	.949	.952	.995	.999	.999	1.000

COMBINED EFFECT OF 2 EXECUTIONS IN SMOKE:

TIME (SEC)	CL (G/M**2)	TRANSMISSION 0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94.GHZ
5.	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10.	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15.	.01	.949	.975	.977	.998	.999	.999	1.000
20.	5.17	.000	.000	.000	.374	.764	.768	.995
25.	6.89	.000	.000	.000	.270	.699	.704	.993
30.	5.78	.000	.000	.000	.333	.740	.745	.994
35.	5.21	.000	.000	.000	.371	.763	.767	.995
40.	4.97	.000	.000	.000	.389	.772	.776	.995
45.	4.93	.000	.000	.000	.392	.774	.778	.995
50.	5.02	.000	.000	.000	.385	.770	.774	.995
55.	5.18	.000	.000	.000	.374	.764	.763	.995
60.	5.37	.000	.000	.000	.360	.756	.760	.995
65.	5.57	.000	.000	.000	.347	.748	.753	.994
70.	5.75	.000	.000	.000	.335	.741	.746	.994
75.	5.89	.000	.000	.000	.327	.736	.741	.994
80.	5.96	.000	.000	.000	.322	.734	.738	.994
85.	5.94	.000	.000	.000	.323	.734	.739	.994
90.	5.83	.000	.000	.000	.331	.739	.743	.994
95.	5.59	.000	.000	.000	.346	.748	.752	.994
100.	5.21	.000	.000	.000	.372	.763	.767	.995
105.	4.67	.000	.000	.000	.412	.784	.788	.995
110.	3.96	.000	.000	.000	.471	.814	.817	.996
115.	3.05	.000	.001	.002	.560	.853	.856	.997
120.	1.52	.001	.036	.045	.749	.924	.925	.998
125.	.84	.021	.159	.179	.852	.957	.958	.999
130.	.67	.046	.229	.252	.880	.966	.966	.999
135.	.56	.076	.292	.317	.899	.971	.972	.999
140.	.48	.111	.350	.375	.913	.975	.976	1.000
145.	.42	.149	.402	.428	.924	.979	.979	1.000
150.	.37	.187	.449	.474	.933	.981	.982	1.000
155.	.32	.226	.492	.515	.940	.983	.984	1.000
160.	.29	.264	.529	.552	.946	.985	.985	1.000
165.	.26	.300	.563	.585	.951	.986	.987	1.000
170.	.24	.335	.593	.614	.956	.988	.988	1.000
175.	.22	.368	.621	.641	.959	.989	.989	1.000
180.	.20	.399	.645	.664	.963	.990	.990	1.000
185.	.08	.696	.841	.851	.985	.996	.996	1.000
190.	.07	.723	.857	.866	.987	.996	.996	1.000
195.	.06	.748	.870	.879	.988	.997	.997	1.000
200.	.06	.770	.882	.890	.989	.997	.997	1.000
205.	.05	.789	.893	.900	.990	.997	.997	1.000
210.	.05	.806	.902	.909	.991	.998	.998	1.000
215.	.04	.822	.911	.916	.992	.998	.998	1.000
220.	.04	.836	.918	.923	.993	.998	.998	1.000
225.	.04	.849	.925	.930	.993	.998	.998	1.000
230.	.03	.860	.931	.935	.994	.998	.998	1.000
235.	.03	.871	.936	.940	.994	.998	.998	1.000
240.	.03	.880	.941	.945	.995	.999	.999	1.000
245.	.03	.888	.945	.949	.995	.999	.999	1.000
250.	.02	.896	.949	.952	.995	.999	.999	1.000

***TRANSMISSION RETURNED TO MAIN FOR WAVELENGTH OF 10.600 MICROMETERS IS .999 AT TIME 250.

DIRT TRANSMISSION MODULE

```
            DIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION    CALCULATION
           *** NOTE -- ALL UNITS ARE MKS UNLESS OTHERWISE SPECIFIED ***

                      PASQUILL CATEGORY              D
            HT     2.00  TEMP   278.30     HT     2.00  WIND     3.60
                         WIND DIRECTION             45.00

                         LATITUDE                  53.00
                    THE INVERSION LAYER HEIGHT IS GROWING

                               SOIL-2
                           SILT CONTENT      .15
                          DEPTH OF SOD          .00

                     30 DEGREE TILTED TIP AT 0.3 METER DEPTH
                    WEIGHT OF CHARGE              ,     6.80 KG.
                     DETONATION DEPTH                    .00

       SIMULTANEOUS BURST , UNIFORMLY DISTRIBUTED CHARGES IN A PARALLELOGRAM
TOTAL NUMBER OF CHARGES IS ,    1  WITH REFERENCE CHARGE AT (      .00,      .00)
          1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (      .00,      .00)
          1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (      .00,      .00)

                    ESTIMATED INVERSION HEIGHT        935

                     TIME AFTER BLAST               1.00

                  WAVELENGTH           10.60 MICROMETERS
           TRANSMITTER COORDINATES      -200.00       .00       2.00
           RECEIVER     COORDINATES      200.00       .00       2.00
                TRANSMITTANCE ALONG THE LINE OF SIGHT   .767-002
                              AERODYNAMIC CLOUD DIMENSIONS

              OBSERVER    COORDINATES          200.00        .00
            THE HEIGHT OF THE CLOUD IS                19.74 METERS
           THE CENTROID COORDINATES ARE                1.34       10.55
           THE WIDTH AT THE CENTROID IS               20.62 METERS
            THE WIDTH AT     2.00 METERS IS           16.56 METERS
                      6  CONTOUR POINTS HAVE BEEN DETERMINED
                                    -5.670       2.000
                                    -8.973      10.550
                                     1.337      19.745
                                     1.337      19.745
                                    11.646      10.550
                                    10.893       2.000

                    ESTIMATED INVERSION HEIGHT        935

                     TIME AFTER BLAST               3.00

                  WAVELENGTH           10.60 MICROMETERS
           TRANSMITTER COORDINATES      -200.00       .00       2.00
           RECEIVER     COORDINATES      200.00       .00       2.00
                TRANSMITTANCE ALONG THE LINE OF SIGHT   .620+000
                              AERODYNAMIC CLOUD DIMENSIONS

              OBSERVER    COORDINATES          200.00        .00
            THE HEIGHT OF THE CLOUD IS                21.89 METERS
           THE CENTROID COORDINATES ARE                8.02       10.55
           THE WIDTH AT THE CENTROID IS               31.26 METERS
            THE WIDTH AT     2.00 METERS IS           21.88 METERS
                      6  CONTOUR POINTS HAVE BEEN DETERMINED
                                    -2.947       2.000
                                    -7.608      10.550
```

```
                   8.020      21.888
                   8.020      21.888
                  23.649      10.550
                  18.928       2.000

     ESTIMATED INVERSION HEIGHT        935

      TIME AFTER BLAST               5.00

         WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00        .00        2.00
RECEIVER    COORDINATES         200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .940+000
                 AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER    COORDINATES           200.00        .00
 THE HEIGHT OF THE CLOUD IS                  23.56 METERS
THE CENTROID COORDINATES ARE                 14.70      10.55
THE WIDTH AT THE CENTROID IS                 38.28 METERS
 THE WIDTH AT     2.00 METERS IS             29.37 METERS
         6  CONTOUR POINTS HAVE BEEN DETERMINED
                   -.224       2.000
                  -4.436      10.550
                  14.704      23.558
                  14.704      23.558
                  33.844      10.550
                  29.151       2.000

     ESTIMATED INVERSION HEIGHT        935

      TIME AFTER BLAST               7.00

         WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00        .00        2.00
RECEIVER    COORDINATES         200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .987+000
                 AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER    COORDINATES           200.00        .00
 THE HEIGHT OF THE CLOUD IS                  24.95 METERS
THE CENTROID COORDINATES ARE                 21.39      10.55
THE WIDTH AT THE CENTROID IS                 43.76 METERS
 THE WIDTH AT     2.00 METERS IS             35.94 METERS
         6  CONTOUR POINTS HAVE BEEN DETERMINED
                   3.436       2.000
                   -.491      10.550
                  21.398      24.948
                  21.388      24.948
                  43.267      10.550
                  39.374       2.000

     ESTIMATED INVERSION HEIGHT        935

      TIME AFTER BLAST               9.00

         WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00        .00        2.00
RECEIVER    COORDINATES         200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .997+000
                 AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER    COORDINATES           200.00        .00
 THE HEIGHT OF THE CLOUD IS                  26.15 METERS
THE CENTROID COORDINATES ARE                 28.07      10.55
THE WIDTH AT THE CENTROID IS                 48.33 METERS
 THE WIDTH AT     2.00 METERS IS             41.56 METERS
         6  CONTOUR POINTS HAVE BEEN DETERMINED
                   7.097       2.000
```

```
                 3.906       10.550
                28.072       26.148
                28.072       26.148
                52.237       10.550
                48.659        2.000

      ESTIMATED INVERSION HEIGHT          935

       TIME AFTER BLAST               11.00

      WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00        .00        2.00
RECEIVER    COORDINATES        200.00        .00        2.00
    TRANSMITTANCE ALONG THE LINE OF SIGHT   .999+000
                   AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER    COORDINATES           200.00        .00
  THE HEIGHT OF THE CLOUD IS                   27.21 METERS
 THE CENTROID COORDINATES ARE                 34.76       10.55
 THE WIDTH AT THE CENTROID IS                 52.30 METERS
  THE WIDTH AT    2.00 METERS IS               46.56 METERS
           6  CONTOUR POINTS HAVE BEEN DETERMINED
                11.382        2.000
                 8.607       10.550
                34.755       27.213
                34.755       27.213
                60.904       10.550
                57.944        2.000

      ESTIMATED INVERSION HEIGHT          935

       TIME AFTER BLAST               13.00

      WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00        .00        2.00
RECEIVER    COORDINATES        200.00        .00        2.00
    TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
                   AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER    COORDINATES           200.00        .00
  THE HEIGHT OF THE CLOUD IS                   28.17 METERS
 THE CENTROID COORDINATES ARE                 41.44       10.55
 THE WIDTH AT THE CENTROID IS                 55.82 METERS
  THE WIDTH AT    2.00 METERS IS               50.94 METERS
           6  CONTOUR POINTS HAVE BEEN DETERMINED
                15.667        2.000
                13.529       10.550
                41.439       28.173
                41.439       28.173
                69.349       10.550
                66.605        2.000

      ESTIMATED INVERSION HEIGHT          935

       TIME AFTER BLAST               15.00

      WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00        .00        2.00
RECEIVER    COORDINATES        200.00        .00        2.00
    TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
                   AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER    COORDINATES           200.00        .00
  THE HEIGHT OF THE CLOUD IS                   29.05 METERS
 THE CENTROID COORDINATES ARE                 48.12       10.55
 THE WIDTH AT THE CENTROID IS                 59.00 METERS
  THE WIDTH AT    2.00 METERS IS               55.00 METERS
           6  CONTOUR POINTS HAVE BEEN DETERMINED
```

```
            20.265        2.000
            18.622       10.550
            48.123       29.050
            48.123       29.050
            77.624       10.550
            75.265        2.000

ESTIMATED INVERSION HEIGHT         935

TIME AFTER BLAST              17.00

       WAVELENGTH           10.60 MICROMETERS
TRANSMITTER COORDINATES     -200.00        .00        2.00
RECEIVER    COORDINATES      200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
               AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER     COORDINATES            200.00        .00
  THE HEIGHT OF THE CLOUD IS                  29.86 METERS
 THE CENTROID COORDINATES ARE               54.81       10.55
 THE WIDTH AT THE CENTROID IS               61.91 METERS
  THE WIDTH AT     2.00 METERS IS             59.38 METERS
           6  CONTOUR POINTS HAVE BEEN DETERMINED
            24.551        2.000
            23.850       10.550
            54.807       29.859
            54.807       29.859
            85.763       10.550
            83.926        2.000

ESTIMATED INVERSION HEIGHT         935

TIME AFTER BLAST              19.00

       WAVELENGTH           10.60 MICROMETERS
TRANSMITTER COORDINATES     -200.00        .00        2.00
RECEIVER    COORDINATES      200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
               AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER     COORDINATES            200.00        .00
  THE HEIGHT OF THE CLOUD IS                  30.61 METERS
 THE CENTROID COORDINATES ARE               61.49       10.55
 THE WIDTH AT THE CENTROID IS               64.60 METERS
  THE WIDTH AT     2.00 METERS IS             63.12 METERS
           6  CONTOUR POINTS HAVE BEEN DETERMINED
            29.148        2.000
            29.190       10.550
            61.490       30.611
            61.490       30.611
            93.791       10.550
            92.273        2.000

ESTIMATED INVERSION HEIGHT         935

TIME AFTER BLAST              21.00

       WAVELENGTH           10.60 MICROMETERS
TRANSMITTER COORDINATES     -200.00        .00        2.00
RECEIVER    COORDINATES      200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
               AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER     COORDINATES            200.00        .00
  THE HEIGHT OF THE CLOUD IS                  31.32 METERS
 THE CENTROID COORDINATES ARE               68.17       10.55
 THE WIDTH AT THE CENTROID IS               67.10 METERS
  THE WIDTH AT     2.00 METERS IS             66.87 METERS
```

```
        6  CONTOUR POINTS HAVE BEEN DETERMINED
                            33.434       2.000
                            34.623      10.550
                            68.174      31.315
                            68.174      31.315
                           101.725      10.550
                           100.309       2.000

       ESTIMATED INVERSION HEIGHT          935

        TIME AFTER BLAST              23.00

       WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00        .00        2.00
RECEIVER    COORDINATES        200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
                    AERODYNAMIC CLOUD DIMENSIONS

    OBSERVER    COORDINATES          200.00        .00
  THE HEIGHT OF THE CLOUD IS                    31.98 METERS
 THE CENTROID COORDINATES ARE                74.86       10.55
 THE WIDTH AT THE CENTROID IS                69.44 METERS
  THE WIDTH AT     2.00 METERS IS               70.31 METERS
        6  CONTOUR POINTS HAVE BEEN DETERMINED
                            38.032       2.000
                            40.135      10.550
                            74.858      31.977
                            74.858      31.977
                           109.580      10.550
                           108.344       2.000

       ESTIMATED INVERSION HEIGHT          935

        TIME AFTER BLAST              25.00

       WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00        .00        2.00
RECEIVER    COORDINATES        200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
                    AERODYNAMIC CLOUD DIMENSIONS

    OBSERVER    COORDINATES          200.00        .00
  THE HEIGHT OF THE CLOUD IS                    32.60 METERS
 THE CENTROID COORDINATES ARE                81.54       10.55
 THE WIDTH AT THE CENTROID IS                71.65 METERS
  THE WIDTH AT     2.00 METERS IS               73.75 METERS
        6  CONTOUR POINTS HAVE BEEN DETERMINED
                            42.630       2.000
                            45.717      10.550
                            81.541      32.603
                            81.541      32.603
                           117.366      10.550
                           116.380       2.000

       ESTIMATED INVERSION HEIGHT          935

        TIME AFTER BLAST              27.00

       WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00        .00        2.00
RECEIVER    COORDINATES        200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
                    AERODYNAMIC CLOUD DIMENSIONS

    OBSERVER    COORDINATES          200.00        .00
  THE HEIGHT OF THE CLOUD IS                    33.20 METERS
 THE CENTROID COORDINATES ARE                88.23       10.55
 THE WIDTH AT THE CENTROID IS                73.73 METERS
```

```
 THE WIDTH AT     2.00 METERS IS            77.19 METERS
         6  CONTOUR POINTS HAVE BEEN DETERMINED
                        47.227       2.000
                        51.359      10.550
                        88.225      33.196
                        88.225      33.196
                       125.091      10.550
                       124.415       2.000

        ESTIMATED INVERSION HEIGHT          935

         TIME AFTER BLAST              29.00

        WAVELENGTH             10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00        .00       2.00
RECEIVER    COORDINATES         200.00        .00       2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001

                   AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER     COORDINATES          200.00        .00
 THE HEIGHT OF THE CLOUD IS                  33.76 METERS
THE CENTROID COORDINATES ARE               94.91      10.55
THE WIDTH AT THE CENTROID IS               75.71 METERS
 THE WIDTH AT     2.00 METERS IS            80.94 METERS
         6  CONTOUR POINTS HAVE BEEN DETERMINED
                        51.513       2.000
                        57.055      10.550
                        94.909      33.761
                        94.909      33.761
                       132.763      10.550
                       132.450       2.000

        ESTIMATED INVERSION HEIGHT          935

         TIME AFTER BLAST              31.00

        WAVELENGTH             10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00        .00       2.00
RECEIVER    COORDINATES         200.00        .00       2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001

                   AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER     COORDINATES          200.00        .00
 THE HEIGHT OF THE CLOUD IS                  34.30 METERS
THE CENTROID COORDINATES ARE              101.59      10.55
THE WIDTH AT THE CENTROID IS               77.59 METERS
 THE WIDTH AT     2.00 METERS IS            84.06 METERS
         6  CONTOUR POINTS HAVE BEEN DETERMINED
                        56.111       2.000
                        62.799      10.550
                       101.593      34.299
                       101.593      34.299
                       140.387      10.550
                       140.173       2.000

        ESTIMATED INVERSION HEIGHT          935

         TIME AFTER BLAST              33.00

        WAVELENGTH             10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00        .00       2.00
RECEIVER    COORDINATES         200.00        .00       2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001

                   AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER     COORDINATES          200.00        .00
 THE HEIGHT OF THE CLOUD IS                  34.81 METERS
THE CENTROID COORDINATES ARE              108.28      10.55
```

```
 THE WIDTH AT THE CENTROID IS              79.38 METERS
  THE WIDTH AT     2.00 METERS IS           87.19 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                        60.708         2.000
                        68.585        10.550
                       108.276        34.814
                       108.276        34.814
                       147.968        10.550
                       147.896         2 000

          ESTIMATED INVERSION HEIGHT          935

           TIME AFTER BLAST               35.00

          WAVELENGTH          10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00          .00        2.00
RECEIVER    COORDINATES         200.00          .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT    .100+001
                     AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER    COORDINATES           200.00          .00
  THE HEIGHT OF THE CLOUD IS                 35.31 METERS
 THE CENTROID COORDINATES ARE              114.96       10.55
 THE WIDTH AT THE CENTROID IS               81.10 METERS
  THE WIDTH AT     2.00 METERS IS           90.31 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                        65.306         2.000
                        74.410        10.550
                       114.960        35.308
                       114.960        35.308
                       155.510        10.550
                       155.619         2.000

          ESTIMATED INVERSION HEIGHT          935

           TIME AFTER BLAST               37.00

          WAVELENGTH          10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00          .00        2.00
RECEIVER    COORDINATES         200.00          .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT    .100+001
                     AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER    COORDINATES           200.00          .00
  THE HEIGHT OF THE CLOUD IS                 35.78 METERS
 THE CENTROID COORDINATES ARE              121.64       10.55
 THE WIDTH AT THE CENTROID IS               82.75 METERS
  THE WIDTH AT     2.00 METERS IS           93.75 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                        69.592         2.000
                        80.271        10.550
                       121.644        35.782
                       121.644        35.782
                       163.017        10.550
                       163.342         2.000

          ESTIMATED INVERSION HEIGHT          935

           TIME AFTER BLAST               39.00

          WAVELENGTH          10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00          .00        2.00
RECEIVER    COORDINATES         200.00          .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT    .100+001
                     AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER    COORDINATES           200.00          .00
  THE HEIGHT OF THE CLOUD IS                 36.24 METERS
```

```
THE CENTROID COORDINATES ARE              128.33      10.55
THE WIDTH AT THE CENTROID IS               84.33 METERS
 THE WIDTH AT     2.00 METERS IS            96.88 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                          74.190       2.000
                          86.164      10.550
                         128.328      36.238
                         128.328      36.238
                         170.492      10.550
                         171.065       2.000

         ESTIMATED INVERSION HEIGHT          935

          TIME AFTER BLAST              41.00

         WAVELENGTH              10.60 MICROMETERS
TRANSMITTER COORDINATES         -200.00          .00        2.00
RECEIVER    COORDINATES          200.00          .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT    .100+001
                   AERODYNAMIC CLOUD DIMENSIONS

    OBSERVER     COORDINATES           200.00         .00
  THE HEIGHT OF THE CLOUD IS                 36.68 METERS
 THE CENTROID COORDINATES ARE             135.01      10.55
 THE WIDTH AT THE CENTROID IS              85.85 METERS
  THE WIDTH AT    2.00 METERS IS           100.00 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                          78.787       2.000
                          92.086      10.550
                         135.011      36.677
                         135.011      36.677
                         177.936      10.550
                         178.787       2.000

         ESTIMATED INVERSION HEIGHT          935

          TIME AFTER BLAST              43.00

         WAVELENGTH              10.60 MICROMETERS
TRANSMITTER COORDINATES         -200.00          .00        2.00
RECEIVER    COORDINATES          200.00          .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT    .100+001
                   AERODYNAMIC CLOUD DIMENSIONS

    OBSERVER     COORDINATES           200.00         .00
  THE HEIGHT OF THE CLOUD IS                 37.10 METERS
 THE CENTROID COORDINATES ARE             141.70      10.55
 THE WIDTH AT THE CENTROID IS              87.32 METERS
  THE WIDTH AT    2.00 METERS IS           102.81 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                          83.385       2.000
                          98.037      10.550
                         141.695      37.102
                         141.695      37.102
                         185.353      10.550
                         186.198       2.000

         ESTIMATED INVERSION HEIGHT          935

          TIME AFTER BLAST              45.00

         WAVELENGTH              10.60 MICROMETERS
TRANSMITTER COORDINATES         -200.00          .00        2.00
RECEIVER    COORDINATES          200.00          .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT    .100+001
                   AERODYNAMIC CLOUD DIMENSIONS

    OBSERVER     COORDINATES           200.00         .00
```

THE HEIGHT OF THE CLOUD IS 37.51 METERS
THE CENTROID COORDINATES ARE 148.38 10.55
THE WIDTH AT THE CENTROID IS 88.73 METERS
THE WIDTH AT 2.00 METERS IS 105.94 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

87.983	2.000
104.013	10.550
148.379	37.512
148.379	37.512
192.745	10.550
193.921	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 47.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 37.91 METERS
THE CENTROID COORDINATES ARE 155.06 10.55
THE WIDTH AT THE CENTROID IS 90.10 METERS
THE WIDTH AT 2.00 METERS IS 108.75 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

92.581	2.000
110.012	10.550
155.063	37.909
155.063	37.909
200.113	10.550
201.331	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 49.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 38.29 METERS
THE CENTROID COORDINATES ARE 161.75 10.55
THE WIDTH AT THE CENTROID IS 91.43 METERS
THE WIDTH AT 2.00 METERS IS 111.87 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

97.175	2.000
116.034	10.550
161.746	38.293
161.746	38.293
207.459	10.550
209.054	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 51.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

```
  OBSERVER     COORDINATES          200.00        .00
 THE HEIGHT OF THE CLOUD IS                 38.67 METERS
THE CENTROID COORDINATES ARE           168.43       10.55
THE WIDTH AT THE CENTROID IS              92.71 METERS
 THE WIDTH AT     2.00 METERS IS          114.69 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                    101.777      2.000
                    122.076     10.550
                    168.430     38.665
                    168.430     38.665
                    214.784     10.550
                    216.464      2.000

        ESTIMATED INVERSION HEIGHT        935

         TIME AFTER BLAST            53.00

        WAVELENGTH          10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00       .00        2.00
RECEIVER    COORDINATES        200.00       .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001

                  AERODYNAMIC CLOUD DIMENSIONS
  OBSERVER     COORDINATES          200.00        .00
 THE HEIGHT OF THE CLOUD IS                 39.03 METERS
THE CENTROID COORDINATES ARE           175.11       10.55
THE WIDTH AT THE CENTROID IS              93.95 METERS
 THE WIDTH AT     2.00 METERS IS          117.50 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                    106.375      2.000
                    128.138     10.550
                    175.114     39.027
                    175.114     39.027
                    222.090     10.550
                    223.875      2.000

        ESTIMATED INVERSION HEIGHT        935

         TIME AFTER BLAST            55.00

        WAVELENGTH          10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00       .00        2.00
RECEIVER    COORDINATES        200.00       .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001

                  AERODYNAMIC CLOUD DIMENSIONS
  OBSERVER     COORDINATES          200.00        .00
 THE HEIGHT OF THE CLOUD IS                 39.38 METERS
THE CENTROID COORDINATES ARE           181.80       10.55
THE WIDTH AT THE CENTROID IS              95.16 METERS
 THE WIDTH AT     2.00 METERS IS          120.00 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                    111.285      2.000
                    134.217     10.550
                    181.797     39.378
                    181.797     39.378
                    229.377     10.550
                    231.285      2.000

        ESTIMATED INVERSION HEIGHT        935

         TIME AFTER BLAST            57.00

        WAVELENGTH          10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00       .00        2.00
RECEIVER    COORDINATES        200.00       .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001

                  AERODYNAMIC CLOUD DIMENSIONS
```

```
    OBSERVER     COORDINATES            200.00        .00
  THE HEIGHT OF THE CLOUD IS                  39.72 METERS
 THE CENTROID COORDINATES ARE               188.48       10.55
 THE WIDTH AT THE CENTROID IS                 96.33 METERS
  THE WIDTH AT     2.00 METERS IS            122.81 METERS
            6  CONTOUR POINTS HAVE BEEN DETERMINED
                    115.883       2.000
                    140.315      10.550
                    188.481      39.720
                    188.481      39.720
                    236.648      10.550
                    238.695       2.000

          ESTIMATED INVERSION HEIGHT         935

           TIME AFTER BLAST              59.00

          WAVELENGTH           10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00        .00        2.00
RECEIVER    COORDINATES         200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT    .100+001
                    AERODYNAMIC CLOUD DIMENSIONS
    OBSERVER     COORDINATES            200.00        .00
  THE HEIGHT OF THE CLOUD IS                  40.05 METERS
 THE CENTROID COORDINATES ARE               195.16       10.55
 THE WIDTH AT THE CENTROID IS                 97.47 METERS
  THE WIDTH AT     2.00 METERS IS            125.62 METERS
            6  CONTOUR POINTS HAVE BEEN DETERMINED
                    120.481       2.000
                    146.428      10.550
                    195.165      40.052
                    195.165      40.052
                    243.902      10.550
                    246.106       2.000

          ESTIMATED INVERSION HEIGHT         935

           TIME AFTER BLAST              61.00

          WAVELENGTH           10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00        .00        2.00
RECEIVER    COORDINATES         200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT    .100+001
                    AERODYNAMIC CLOUD DIMENSIONS
    OBSERVER     COORDINATES            200.00        .00
  THE HEIGHT OF THE CLOUD IS                  40.38 METERS
 THE CENTROID COORDINATES ARE               201.85       10.55
 THE WIDTH AT THE CENTROID IS                 98.58 METERS
  THE WIDTH AT     2.00 METERS IS            128.12 METERS
            6  CONTOUR POINTS HAVE BEEN DETERMINED
                    125.391       2.000
                    152.557      10.550
                    201.849      40.376
                    201.849      40.376
                    251.140      10.550
                    253.516       2.000

          ESTIMATED INVERSION HEIGHT         935

           TIME AFTER BLAST              63.00

          WAVELENGTH           10.60 MICROMETERS
TRANSMITTER COORDINATES        -200.00        .00        2.00
RECEIVER    COORDINATES         200.00        .00        2.00
     TRANSMITTANCE ALONG THE LINE OF SIGHT    .100+001
```

```
                    AERODYNAMIC CLOUD DIMENSIONS
 OBSERVER      COORDINATES              200.00          .00
 THE HEIGHT OF THE CLOUD IS                     40.69 METERS
THE CENTROID COORDINATES ARE                  208.53       10.55
THE WIDTH AT THE CENTROID IS                   99.66 METERS
 THE WIDTH AT     2.00 METERS IS               130.62 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                         130.301       2.000
                         158.700      10.550
                         208.532      40.691
                         208.532      40.691
                         258.364      10.550
                         260.926       2.000

        ESTIMATED INVERSION HEIGHT          935

          TIME AFTER BLAST               65.00

          WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES          -200.00         .00         2.00
RECEIVER     COORDINATES          200.00         .00         2.00
      TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
                    AERODYNAMIC CLOUD DIMENSIONS
 OBSERVER      COORDINATES              200.00          .00
 THE HEIGHT OF THE CLOUD IS                     41.00 METERS
THE CENTROID COORDINATES ARE                  215.22       10.55
THE WIDTH AT THE CENTROID IS                  100.72 METERS
 THE WIDTH AT     2.00 METERS IS               133.12 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                         135.212       2.000
                         164.857      10.550
                         215.216      40.998
                         215.216      40.998
                         265.575      10.550
                         268.337       2.000

        ESTIMATED INVERSION HEIGHT          935

          TIME AFTER BLAST               67.00

          WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES          -200.00         .00         2.00
RECEIVER     COORDINATES          200.00         .00         2.00
      TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
                    AERODYNAMIC CLOUD DIMENSIONS
 OBSERVER      COORDINATES              200.00          .00
 THE HEIGHT OF THE CLOUD IS                     41.30 METERS
THE CENTROID COORDINATES ARE                  221.90       10.55
THE WIDTH AT THE CENTROID IS                  101.74 METERS
 THE WIDTH AT     2.00 METERS IS               135.00 METERS
          6  CONTOUR POINTS HAVE BEEN DETERMINED
                         140.435       2.000
                         171.028      10.550
                         221.900      41.298
                         221.900      41.298
                         272.772      10.550
                         275.435       2.000

        ESTIMATED INVERSION HEIGHT          935

          TIME AFTER BLAST               69.00

          WAVELENGTH            10.60 MICROMETERS
TRANSMITTER COORDINATES          -200.00         .00         2.00
RECEIVER     COORDINATES          200.00         .00         2.00
      TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
```

```
                    AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER     COORDINATES           200.00        .00
  THE HEIGHT OF THE CLOUD IS                  41.59 METERS
 THE CENTROID COORDINATES ARE             228.58        10.55
 THE WIDTH AT THE CENTROID IS             102.75 METERS
  THE WIDTH AT     2.00 METERS IS           137.50 METERS
            6  CONTOUR POINTS HAVE BEEN DETERMINED
                          145.345       2.000
                          177.211      10.550
                          228.584      41.591
                          228.584      41.591
                          279.956      10.550
                          282.845       2.000

          ESTIMATED INVERSION HEIGHT         935

           TIME AFTER BLAST               71.00

          WAVELENGTH          10.60 MICROMETERS
TRANSMITTER COORDINATES       -200.00         .00        2.00
RECEIVER     COORDINATES       200.00         .00        2.00
      TRANSMITTANCE ALONG THE LINE OF SIGHT   .100+001
                    AERODYNAMIC CLOUD DIMENSIONS

   OBSERVER     COORDINATES           200.00        .00
  THE HEIGHT OF THE CLOUD IS                  41.88 METERS
 THE CENTROID COORDINATES ARE             235.27        10.55
 THE WIDTH AT THE CENTROID IS             103.72 METERS
  THE WIDTH AT     2.00 METERS IS           139.69 METERS
            6  CONTOUR POINTS HAVE BEEN DETERMINED
                          150.568       2.000
                          183.406      10.550
                          235.267      41.877
                          235.267      41.877
                          287.129      10.550
                          290.255       2.000
```

LASER TRANSMITTANCE MODULE

*** WARNING INPUT WAVELENGTH 10.600 CHANGED TO 10.591 NEAREST STANDARD WAVELENGTH ***

WAVELENGTH (MICRONS)	H2O PRESSURE (TORR)	TEMPERATURE (ABS)	ABSORPTION COEFFICIENT (KM-1)	LINE	PATHLENGTH (KM)	TRANSMISSION
10.591	5.821	278.46	.111+000	P(20)	.4000+000	.9566+000

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .719 KM
TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .481 KM
TOTAL OPTICAL DEPTH = 25.11

TRANSMITTANCE ALONG LINE-OF-SIGHT = .12449-010

SEEKER COORDINATES (KM) XSEEKER	YSEEKER	ZSEEKER	TARGET COORDINATES (KM) XTARGET	YTARGET	ZTARGET
.200	.000	.600	-.200	.000	.002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT INTERSECTION COORDINATES (KM) XUPPER	YUPPER	ZUPPER	XLOWER	YLOWER	ZLOWER
ST/ 1	.200	.000	.600	-.068	.000	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	25.11	.12449-010

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .599 KM

TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .361 KM

TOTAL OPTICAL DEPTH = 16.57

TRANSMITTANCE ALONG LINE-OF-SIGHT = .63594-007

SEEKER COORDINATES (KM) XSEEKER	YSEEKER	ZSEEKER	TARGET COORDINATES (KM) XTARGET	YTARGET	ZTARGET
.133	.000	.500	-.200	.000	.002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT INTERSECTION COORDINATES (KM) XUPPER	YUPPER	ZUPPER	XLOWER	YLOWER	ZLOWER
ST/ 1	.133	.000	.500	-.068	.000	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	16.57	.63594-007

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .359 KM
TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .120 KM
TOTAL OPTICAL DEPTH = 3.02

TRANSMITTANCE ALONG LINE-OF-SIGHT = .48846-001

SEEKER COORDINATES (KM)			TARGET COORDINATES (KM)		
XSEEKER	YSEEKER	ZSEEKER	XTARGET	YTARGET	ZTARGET
.000	.000	.300	-.200	.000	.002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT INTERSECTION COORDINATES (KM)					
	XUPPER	YUPPER	ZUPPER	XLOWER	YLOWER	ZLOWER
ST/ 1	.000	.000	.300	-.067	.000	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	3.02	.48846-001

INVERSE STATIC TARGET DETECTION MODEL

DEVICE NUMBER	8	TARGET INTRINSIC CONTRAST OR TEMPERATURE DIFFERENCE	1.000
FOV TYPE -	WIDE	MINIMUM TARGET DIMENSION (M)	2.300
FOV (DEG)	10.620	ACQUISITION LEVEL (50 PCNT)	1.000
MAGNIFICATION	1.000 (ARBITRARY DEFAULT)	SEARCH ZONE (DEGREES**2)	81.000

FOR NO GREATER INPUT DETECTION PROB.	TIME(SEC)	UNDER INPUT AMB. ILLUM. (FT CDLS)	AND INPUT SKY/GROUND RATIO	AT INPUT RANGE (KM)	REQUIRES (TO DEFEAT DEVICE) AT MOST: COMPUTED RESOLVABLE CYCLES, RC	CONTRAST OR TEMP. DIFF. AT DEVICE	COMPUTED TOTAL PATH TRANSMITTANCE	COMMENTS
.990	1.32	10.000000	2.000	.24	.773	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.
.900	1.58	10.000000	2.000	.24	.567	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.
.750	2.16	10.000000	2.000	.24	.446	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.
.500	3.89	10.000000	2.000	.24	.313	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.

** CONTINUED ON NEXT PAGE **

INVERSE STATIC TARGET DETECTION MODEL

DEVICE NUMBER	8	TARGET INTRINSIC CONTRAST OR TEMPERATURE DIFFERENCE	1.000
FOV TYPE -	WIDE	MINIMUM TARGET DIMENSION (M)	2.300
FOV (DEG)	10.620	ACQUISITION LEVEL (50 PCNT)	1.000
MAGNIFICATION	1.000 (ARBITRARY DEFAULT)	SEARCH ZONE (DEGREES**2)	81.000

FOR NO GREATER INPUT DETECTION PROB.	TIME(SEC)	UNDER INPUT AMB. ILLUM. (FT CDLS)	AND INPUT SKY/GROUND RATIO	AT INPUT RANGE (KM)	REQUIRES (TO DEFEAT DEVICE) AT MOST: COMPUTED RESOLVABLE CYCLES, RC	CONTRAST OR TEMP. DIFF. AT DEVICE	COMPUTED TOTAL PATH TRANSMITTANCE	COMMENTS
.100	24.64	10.000000	2.000	.24	.059	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.

*** FINAL TOTAL TRANSMISSION FROM ITAM = .037

```
***********************************
*                                 *
*    CWIC MUNITION EXPENDITURES   *
*                                 *
***********************************


         MUNITION EXPENDITURES
          FOR HC AND WP SMOKE

SLANT RANGE OBS-TGT    - KM       =   .400
ELEVATION OF TARGET    - DEG      =    .00
AZIMUTH OF TARGET      - DEG      =  90.00
AVG ROUGHNESS ELEMENT  - CM       =   74.0

ATMOSPHERIC EXTINCTION CORRECTIONS
CORRECTED FOR VISIBILITY   -           YES
MARITIME ARCTIC AIR MASS   -           YES
MARITIME POLAR AIR MASS    -           NO
CONTINENTAL POLAR AIR MASS -           NO
CORRECTED FOR RAIN       -             NO
CORRECTED FOR SNOW       -             NO

METEOROLOGICAL INPUTS
WINDSPEED              - M/SEC    =   3.60
WIND DIRECTION         - DEG      = 225.00
PASQUILL CATEGORY      -          =      D
VISIBILITY             - KM       =  9.302
RELATIVE HUMIDITY      - PERCENT  =   87.1

TRANSMISSION THRESHOLDS   TOTAL      SMOKE
VISIBLE:                   .037       .043
NEAR IR:                   .037       .041
MID IR:                    .037       .044
FAR IR:                    .037       .039
```

- VISIBLE: -

SCREEN	LENGTH METERS	DURATION MINUTES
	400.	15.00

HC SMOKE SCREEN

105MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	8.		51.	
SUSTAINING:	8.	.5	51.	60.

155MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	3.		162.	
SUSTAINING:	2.	.5	328.	16.

WP SMOKE SCREEN

105MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	6.		82.	
SUSTAINING:	6.	.8	82.	129.

155MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	3.		246.	
SUSTAINING:	3.	.8	246.	31.

- NEAR IR: -

SCREEN	LENGTH METERS	DURATION MINUTES
	400.	15.00

HC SMOKE SCREEN

105MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	23.		18.	
SUSTAINING:	23.	.5	18.	173.

155MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	4.		114.	
SUSTAINING:	4.	.5	114.	30.

WP SMOKE SCREEN

105MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	8.		66.	
SUSTAINING:	8.	.8	66.	172.

155MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	4.		196.	
SUSTAINING:	3.	.8	196.	41.

- MID IR: -

	LENGTH METERS	DURATION MINUTES
SCREEN	400.	15.00

WP SMOKE SCREEN

	ROUNDS/ 60 METERS	RATE/ MINUTE	TOTAL ROUNDS
105MM:	8.	1.	1319.
155MM:	3.	1.	495.

- FAR IR: -

	LENGTH METERS	DURATION MINUTES
SCREEN	400.	15.00

WP SMOKE SCREEN

	ROUNDS/ 60 METERS	RATE/ MINUTE	TOTAL ROUNDS
105MM:	6.	1.	989.
155MM:	2.	1.	330.

OVERCAST SKY RADIATIVE TRANSFER MODULE

-- RADIATION UNDER OVERCAST SKY --

```
XO     =      -.067 (KM)       XT  =      -.200 (KM)
YO     =       .000            YT  =       .000
ZO     =       .200            ZT  =       .002
ZC     =       .200            LC  = 2.0000+002 (W/M2-SR-MU)
LAMBDA =     10.600 (MU)       LG  = 5.0000+001
TEMP   =      9.8   (DEG.C)    LB0 = 5.0000+001
KAPPA  = 4.0000+001 (KM-1)     W0  =      1.000
ETA    =       .700
```

**THERMAL CALCULATION OF PATH RADIANCE

BBTEMP= 7.4082+000 W/M2-SR-MU

PATH LENGTH (KM)	TRANSMITTANCE	PATH RADIANCE (W/M2-SR-MU)	CONTRAST TRANSMITTANCE
.239	.00007	2.409+001	.00015

-- RADIATION UNDER OVERCAST SKY --

```
XO     =      -.133 (KM)       XT  =      -.200 (KM)
YO     =       .000            YT  =       .000
ZO     =       .100            ZT  =       .002
ZC     =       .200            LC  = 2.0000+002 (W/M2-SR-MU)
LAMBDA =     10.600 (MU)       LG  = 5.0000+001
TEMP   =      9.8   (DEG.C)    LB0 = 5.0000+001
KAPPA  = 4.0000+001 (KM-1)     W0  =      1.000
ETA    =       .700
```

**THERMAL CALCULATION OF PATH RADIANCE

BBTEMP= 7.4082+000 W/M2-SR-MU

PATH LENGTH (KM)	TRANSMITTANCE	PATH RADIANCE (W/M2-SR-MU)	CONTRAST TRANSMITTANCE
.119	.00872	9.826-001	.30742

SELF-SCREENING SMOKE GRENADE MODULE

```
***************************************************************************
*                              PROGRAM GRNADE                             *
*                                EOSAEL80                                 *
***************************************************************************
```

*****CARD INPUT*****

NAME							
OUTP	.000	.000	.000	.000	.000	.000	.000
OBSC	200.000	.000	2.000	.000	.000	.000	.000
MUNC	-200.000	.000	2.000	95.000	100.000	10.000	.000
TARC	-200.000	40.000	2.000	.000	.000	.000	.000
BART	5.000	400.000	5.000	90.000	.000	.000	.000
MUNT	1.000	.793	14.300	1.000	.000	4.700	.070
METR	50.000	2.000	220.000	4.000	20.000	.000	.000
EXTC	.000	.000	.000	.000	.000	.000	.000
BURN	.000	.000	.000	.000	.000	.000	.000
MISC	.000	.000	.000	.000	.000	.000	.000
GO	.000	.000	.000	.000	.000	.000	.000

*****INPUT*****ALL LENGTHS IN METERS

```
METEOROLOGICAL:
   WIND SPEED              3.6 M/S
   WIND DIRECTION        225.0 DEG
   PASQUILL CATEGORY       4
   RELATIVE HUMIDITY      87.1 %
   NOTE: X AXIS HEADING:    90.0 DEG CLOCKWISE FROM NORTH (DCWFN)
OBSERVER/TARGET:              EXTINCTION COEFFICIENTS:
                                   MICRONS    M**2/GM
   X(OBS)      200.0               0.4-0.7     4.304
   Y(OBS)         .0               0.7-1.2     2.166
   Z(OBS)        2.0                 1.06      1.541
   X(TAR)     -200.0               3.0-5.0      .350
   Y(TAR)       40.0               8.0-12.      .338
   Z(TAR)        2.0                 10.6       .364
                                   94.0 GHZ     .001
TANK/MUNITION DATA:                  DIFFUSION PARAMETERS:
   X(TANK)               -200.0          SIGZ(XREF)                         7.2
   Y(TANK)                   .0          XREF                             100.0
   Z(TANK)                  2.0          MIXING HEIGHT(HM)                300.0
   HEADING(DCWFN)          95.0          SCAVENGING COEFF(HK)              .002
   RANGE                  100.0          REFLECTION COEFF(RC)              .700
   NO GRNADES               1.0          SETTLING VELOCITY(VS)       .021 CM/S
   SMOKE MASS(GM)         359.7          VERTICAL WIND EXPONENT(WPOWR)     .000
   LINE LENGTH             10.0          VERTICAL DIFF CONSTANT(ZDIFF)    1.400
   BURN CONSTANT       .070 1/S          CROSSWIND DIFF CONSTANT(YDIFF)    .355
   EFFICIENCY              62.0          YIELD FACTOR                      4.7
```

*****OUTPUT*****

TIME (SEC)	CL (GM/M**2)	TRANSMISSION 0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94GHZ
5.0	.014	.940	.969	.978	.995	.995	.995	1.000
10.0	4.659	.000	.000	.001	.196	.207	.183	.995
15.0	4.320	.000	.000	.001	.220	.232	.208	.996
20.0	3.819	.000	.000	.003	.263	.275	.249	.996
25.0	3.384	.000	.001	.005	.306	.319	.292	.997
30.0	3.016	.000	.001	.010	.348	.361	.334	.997
35.0	2.705	.000	.003	.015	.388	.401	.374	.997
40.0	2.442	.000	.005	.023	.425	.438	.411	.998
45.0	2.218	.000	.008	.033	.460	.472	.446	.998
50.0	2.028	.000	.012	.044	.492	.504	.478	.998
55.0	1.864	.000	.018	.057	.521	.533	.507	.998
60.0	1.722	.001	.024	.070	.547	.559	.534	.998
65.0	1.599	.001	.031	.085	.571	.582	.559	.998
70.0	1.491	.002	.040	.100	.593	.604	.581	.999
75.0	1.396	.002	.049	.116	.613	.624	.602	.999
80.0	1.312	.004	.058	.132	.632	.642	.620	.999
85.0	1.237	.005	.069	.149	.649	.658	.637	.999
90.0	1.170	.006	.079	.165	.664	.673	.653	.999
95.0	1.110	.008	.090	.181	.678	.687	.668	.999
100.0	1.055	.011	.102	.197	.691	.700	.681	.999
105.0	1.006	.013	.113	.212	.703	.712	.693	.999
110.0	.961	.016	.125	.227	.714	.723	.705	.999
115.0	.920	.019	.136	.242	.725	.733	.716	.999
120.0	.882	.022	.148	.257	.734	.742	.725	.999
125.0	.847	.026	.160	.271	.744	.751	.735	.999
130.0	.814	.030	.171	.285	.752	.759	.743	.999
135.0	.785	.034	.183	.298	.760	.767	.752	.999
140.0	.757	.038	.194	.312	.767	.774	.759	.999
145.0	.731	.043	.205	.324	.774	.781	.766	.999
150.0	.707	.048	.216	.337	.781	.788	.773	.999
155.0	.684	.053	.227	.348	.787	.794	.780	.999
160.0	.663	.058	.238	.360	.793	.799	.786	.999
165.0	.643	.063	.248	.371	.799	.805	.791	.999
170.0	.624	.068	.259	.382	.804	.810	.797	.999
175.0	.606	.074	.269	.393	.809	.815	.802	.999
180.0	.590	.079	.279	.403	.814	.819	.807	.999
185.0	.574	.085	.289	.413	.818	.824	.812	.999
190.0	.559	.090	.298	.423	.822	.828	.816	.999
195.0	.545	.096	.307	.432	.826	.832	.820	.999
200.0	.531	.102	.317	.441	.830	.836	.824	.999
205.0	.518	.108	.326	.450	.834	.839	.828	.999
210.0	.506	.113	.334	.459	.838	.843	.832	.999
215.0	.494	.119	.343	.467	.841	.846	.835	1.000
220.0	.483	.125	.351	.475	.844	.849	.839	1.000
225.0	.472	.131	.360	.483	.848	.852	.842	1.000
230.0	.462	.137	.368	.491	.851	.855	.845	1.000
235.0	.452	.143	.375	.498	.854	.858	.848	1.000
240.0	.443	.149	.383	.505	.856	.861	.851	1.000
245.0	.434	.155	.391	.512	.859	.864	.854	1.000
250.0	.425	.160	.398	.519	.862	.866	.857	1.000
255.0	.417	.166	.405	.526	.864	.869	.859	1.000
260.0	.409	.172	.412	.532	.867	.871	.862	1.000
265.0	.401	.178	.419	.539	.869	.873	.864	1.000
270.0	.394	.184	.426	.545	.871	.875	.866	1.000
275.0	.387	.189	.433	.551	.873	.877	.869	1.000
280.0	.380	.195	.439	.557	.876	.880	.871	1.000
285.0	.373	.201	.446	.563	.878	.881	.873	1.000
290.0	.367	.206	.452	.568	.880	.883	.875	1.000
295.0	.361	.212	.458	.574	.881	.885	.877	1.000
300.0	.355	.217	.464	.579	.883	.887	.879	1.000
305.0	.349	.223	.470	.584	.885	.889	.881	1.000
310.0	.343	.228	.475	.589	.887	.890	.883	1.000
315.0	.338	.234	.481	.594	.888	.892	.884	1.000
320.0	.333	.239	.487	.599	.890	.894	.886	1.000
325.0	.327	.244	.492	.604	.892	.895	.888	1.000
330.0	.322	.250	.497	.608	.893	.897	.889	1.000
335.0	.318	.255	.503	.613	.895	.898	.891	1.000
340.0	.313	.260	.508	.617	.896	.900	.892	1.000
345.0	.308	.265	.513	.622	.898	.901	.894	1.000
350.0	.304	.270	.518	.626	.899	.902	.895	1.000
355.0	.300	.275	.522	.630	.900	.904	.897	1.000
360.0	.296	.280	.527	.634	.902	.905	.898	1.000
365.0	.292	.285	.532	.638	.903	.906	.899	1.000
370.0	.288	.290	.536	.642	.904	.907	.901	1.000
375.0	.284	.295	.541	.646	.905	.909	.902	1.000
380.0	.280	.299	.545	.649	.907	.910	.903	1.000

```
 385.0       .277      .304     .549     .653     .908     .911     .904    1.000
 390.0       .273      .309     .554     .657     .909     .912     .905    1.000
 395.0       .270      .313     .558     .660     .910     .913     .907    1.000
 400.0       .266      .318     .562     .664     .911     .914     .908    1.000
 *****CARD INPUT*****
--------------------------------------------------------------------------------
 DONE            .000      .000      .000      .000      .000      .000      .000
*****PROGRAM GRNADE END*****

COMBINED TRANSMISSION FOR THE SELECTED MODULES =  .2331-019
```

RUN NUMBER 2

INDIVIDUAL MODULES SELECTED
BASCAT
FCLOUD

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	9433.962	9433.962
WAVELENGTH(MICRONS)	1.060	1.060
FREQUENCY(GHZ)	283018.863	283018.863

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

```
CLASS  1 = FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS  2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLASS  3 = FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLASS  4 = FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS  5 = DUST WITH VIS LT 3 KM.
CLASS  6 = DUST WITH VIS GE 3 KM.
CLASS  7 = DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM.
CLASS  8 = DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM.
CLASS  9 = DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM.
CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM.
CLASS 11 = SNOW WITH VIS LT 1 KM.
CLASS 12 = SNOW WITH 1 LE VIS LT 3 KM.
CLASS 13 = SNOW WITH 3 LE VIS LT 7 KM.
CLASS 14 = SNOW WITH VIS GE 7 KM.
CLASS 15 = NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 16 = NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 17 = VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 18 = VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLASS 19 = CEILING HEIGHT LT 300 M.
CLASS 20 = CEILING HEIGHT LT 1000 M.
CLASS 21 = NO CEILING.
CLASS 22 = ALL CONDITIONS COMBINED.
```

EOSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS DURING SPRING AT 03-09 (LST).

CLASS NO.	FREQCY CLASS (%)	MEAN TEMP (C)	MEAN DP (C)	MEAN AH (GM/CU.M)	MEAN RH (%)	MEAN VIS (KM)	MEAN PRESS (MB)	MEAN/STDEV WNDVEL (MPS)	MEAN CLDHT (KM)	MEAN/STDEV CLDCVR (%)	FREQCY A (%)	FREQCY B (%)	FREQCY C (%)	FREQCY D (%)	FREQCY E (%)	FREQCY F (%)
1	4.5	3.5	3.2	6.3	98.2	.365	1016.7	1.9/ 1.7	.929	74.4/ 39.2	.0	2.6	12.9	69.2	1.4	14.0
2	8.2	4.1	3.2	6.4	93.8	1.945	1016.1	2.3/ 1.7	1.818	60.0/ 40.4	.0	4.1	19.0	52.6	5.1	19.2
3	17.9	5.1	3.3	6.4	89.0	4.502	1016.4	2.8/ 2.1	2.124	57.8/ 39.4	.0	4.3	15.4	52.4	8.9	19.0
4	9.6	5.8	3.3	6.3	84.5	9.224	1016.0	3.4/ 2.2	1.911	56.2/ 38.6	.0	3.1	12.5	53.1	11.5	19.7
5	.0	.0	.0	.0	.0	.000	.0	.0/ .0	.000	.0/ .0	.0	.0	.0	.0	.0	.0
6	.0	.8	-1.4	5.5	85.5	10.617	1012.3	4.2/ 3.7	1.279	64.0/ 38.2	.0	.0	5.9	64.7	.0	29.4
7	.6	5.9	5.5	7.3	97.6	.460	1013.4	2.9/ 2.6	.204	98.5/ 10.3	.0	.0	.3	97.9	.3	1.7
8	2.7	6.6	5.8	7.5	95.2	1.955	1011.8	3.6/ 2.3	.565	98.4/ 8.7	.0	.1	1.9	95.4	1.6	1.0
9	5.8	6.8	5.6	7.3	92.4	4.590	1010.3	4.5/ 2.7	.872	98.0/ 7.3	.0	.2	2.2	94.5	1.7	1.5
10	7.1	6.9	4.9	7.0	88.0	12.633	1008.8	5.6/ 3.0	1.340	92.7/ 14.5	.0	.1	1.8	92.8	3.5	1.8
11	.2	-1.0	-1.6	4.4	95.9	.522	1009.9	4.3/ 3.2	.428	96.9/ 13.5	.0	.9	2.7	95.5	.0	.9
12	.7	-1.1	-1.8	4.4	95.0	1.870	1009.9	4.4/ 2.7	.270	97.4/ 10.7	.0	.2	2.2	96.7	.2	.7
13	.8	-1.4	-2.7	4.1	91.2	4.436	1013.4	4.6/ 2.7	.824	94.1/ 14.4	.0	.0	2.7	93.0	2.3	2.1
14	.8	-.5	-2.5	4.2	86.9	13.761	1010.9	5.6/ 3.7	1.359	83.8/ 22.9	.0	.0	1.7	91.1	4.5	2.6
15	40.3	5.2	2.3	5.9	82.3	14.756	1015.9	3.9/ 2.6	2.293	51.9/ 37.9	.0	2.1	10.1	54.8	13.4	19.5
16	.8	14.7	12.5	11.1	87.3	12.319	1011.7	3.2/ 1.9	2.938	66.5/ 29.8	.0	6.9	17.7	47.0	12.8	15.7
17	3.6	4.0	3.8	6.6	98.4	.343	1015.5	2.4/ 1.9	.088	99.0/ 4.9	.0	.0	1.8	97.1	.4	.7
18	10.0	4.6	4.0	6.7	96.4	1.312	1014.2	3.1/ 2.2	.189	98.0/ 6.3	.0	.0	2.6	94.9	1.2	1.4
19	14.2	4.9	4.2	6.7	95.3	3.326	1013.7	3.9/ 2.6	.143	98.1/ 6.0	.0	.0	1.2	97.0	1.0	.7
20	35.4	5.2	3.8	6.5	90.9	7.312	1012.9	4.4/ 2.8	.390	94.9/ 9.4	.0	.0	2.0	93.8	2.7	1.6
21	37.2	4.2	1.8	5.8	85.1	10.680	1018.1	2.9/ 2.3	1.878	16.4/ 17.2	.0	5.6	17.9	28.5	13.4	34.6
22	100.0	5.3	3.2	6.3	87.1	9.302	1015.0	3.6/ 2.6	1.885	63.2/ 38.7	.0	2.5	10.7	61.9	9.1	15.8

CLASS NO.	FREQCY CLASS (%)	FREQCY WNDDIR 015 (%)	FREQCY WNDDIR 045 (%)	FREQCY WNDDIR 075 (%)	FREQCY WNDDIR 105 (%)	FREQCY WNDDIR 135 (%)	FREQCY WNDDIR 165 (%)	FREQCY WNDDIR 195 (%)	FREQCY WNDDIR 225 (%)	FREQCY WNDDIR 255 (%)	FREQCY WNDDIR 285 (%)	FREQCY WNDDIR 315 (%)	FREQCY WNDDIR 345 (%)	FREQCY WNDDIR VBL (%)
1	4.5	8.6	11.1	8.3	4.0	2.8	4.6	5.2	6.9	8.2	7.0	4.7	7.2	21.3
2	8.2	8.5	12.0	10.0	5.9	4.6	5.7	5.7	8.7	8.5	5.8	3.9	5.9	14.8
3	17.9	7.6	10.1	11.4	8.3	6.7	7.0	7.7	10.1	8.4	5.3	3.6	5.3	8.5
4	9.6	6.3	7.7	9.8	7.8	7.4	8.6	9.5	11.3	11.1	6.8	3.9	4.7	5.1
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	23.5	11.8	.0	5.9	5.9	11.8	11.8	5.9	5.9	5.9	11.8
7	.6	8.9	11.1	10.0	3.9	1.4	2.5	4.4	12.2	12.2	10.6	7.2	9.4	6.1
8	2.7	6.5	7.8	6.9	4.2	3.3	4.9	8.3	15.6	14.5	9.7	6.8	5.3	6.0
9	5.8	5.3	6.6	4.7	3.6	4.0	6.8	11.3	19.5	14.9	9.5	6.1	5.1	2.5
10	7.1	4.2	4.1	2.6	2.3	3.6	7.3	12.3	21.7	16.9	11.5	7.4	5.0	1.1
11	.2	10.0	12.7	3.6	1.8	1.8	1.8	3.6	3.6	13.6	12.7	11.8	10.9	11.8
12	.7	11.1	12.9	7.3	3.8	3.1	2.9	4.5	8.2	13.8	10.7	7.8	12.0	1.8
13	.8	10.1	11.5	9.5	3.5	1.9	2.7	4.1	9.3	15.4	10.1	7.8	10.7	3.5
14	.8	7.0	6.8	7.4	3.2	.6	1.9	1.9	8.5	21.9	17.5	12.1	8.7	2.6
15	40.3	7.5	9.1	9.7	6.8	4.7	6.8	8.4	13.2	10.5	7.8	5.0	5.9	4.8
16	.8	3.0	4.5	4.3	10.2	10.8	20.7	11.4	14.0	7.3	2.8	2.8	1.8	6.3
17	3.6	10.0	12.9	8.2	3.8	2.5	3.3	5.0	8.8	10.0	8.8	6.1	8.6	12.1
18	10.0	9.3	12.0	8.0	3.6	2.5	3.5	5.4	11.2	12.6	9.6	6.4	7.6	8.3
19	14.2	9.2	10.9	7.5	2.9	1.7	2.7	5.4	12.7	14.8	10.7	7.9	8.3	5.3
20	35.4	7.5	8.2	6.2	2.9	2.1	4.0	7.6	16.1	16.1	11.6	7.3	7.2	3.2
21	37.2	8.1	10.8	12.7	8.7	6.3	7.2	7.1	8.4	6.9	5.0	3.6	5.2	9.9
22	100.0	7.2	8.9	9.0	6.3	5.0	6.8	8.3	12.7	10.9	7.6	4.9	5.7	6.6

VISIBILITY
9.30 KM

BASCAT LASER SCATTERING MODULE

```
***************************************************
*                                                 *
*        MONTE CARLO MULTIPLE SCATTERING          *
*             AEROSOL SCATTERING                  *
*                                                 *
***************************************************

                PARAMETERS FOR THIS RUN

                   WHITE PHOSPHORUS
WAVELENGTH= 1.060 MICROMETERS      ALBEDO= .999
AEROSOL EXTINCTION COEFFICIENT= .4000+001 KM**-1

             ELLIPSOIDAL AEROSOL CLOUD
         COORDINATE ORIGIN AT CENTER OF CLOUD
   Z-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH

                 SOURCE PARAMETERS
SOURCE XYZ COORDINATES(KM)=  -.2000     .0000   -.0980
SOURCE AXIS POLAR ANGLE  = 90.000 DEGREES
SOURCE AXIS AZIMUTH ANGLE=    .000 DEGREES
SOURCE APERTURE RADIUS(MM)= 50.000
SOURCE BEAM SPREAD ANGLE = .1293-004 RADIANS

                DETECTOR PARAMETERS
CONE OF VIEW HALF-ANGLE    =  1.000 DEGREES
DETECTOR APERTURE RADIUS   =  1.000 CM
DETECTOR XYZ COORDINATES(KM)=    .2000     .0000   -.0980
DETECTOR AXIS POLAR ANGLE  = 90.000 DEGREES
DETECTOR AXIS AZIMUTH ANGLE=180.000 DEGREES

              GROUND PLANE PARAMETERS
     ISOTROPIC REFLECTION FROM GROUND PLANE
GROUND PLANE Z-COORDINATE ZG(KM)=  -.100
GROUND PLANE ALBEDO, ALBG,       =   .500

                 CLOUD PARAMETERS
ELLIPSOID PRINCIPAL XYZ HALF-AXES(KM)      =   .1000    .2000    .1000
EULER ANGLES THE,PHE,PSE OF ELLIPSOID(DEG)=    .0000    .0000    .0000
OPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES    =   .8000   1.6000    .8000
```

STEADY STATE POWER TO DETECTOR, FOR UNIT SOURCE POWER

ORDER	STEADY STATE POWER	NUMBER OF PHOTONS
0	.27973-001	.500000+004
1	.66398-009	.500000+004
2	.90441-011	.499900+004
3	.18337-012	.500000+004
4	.12166-013	.500000+004
5	.58910-015	.500000+004
TOTAL	.27973-001	

POWER INTO DETECTOR FOR 1 PULSE(S) OF DIFFERENT LENGTH

PULSE NUMBER 1 HAS LENGTH .2667+000 MICROSECONDS
DETECTOR RESPONSE CUTOFF TIME FOR PULSE NUMBER 1 IS .7067+001 MICROSECONDS

DETECTOR RESPONSE, POWER AS A FUNCTION OF TIME, FOR UNIT PULSE POWER

		POWER FROM EACH ORDER					
TIME	TOTAL POWER	0	1	2	3	4	5
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.1333+000	.2797-001	.2797-001	.6640-009	.9044-011	.1828-012	.1188-013	.2368-015
.2667+000	.2797-001	.2797-001	.6640-009	.9044-011	.1833-012	.1206-013	.4930-015
.4000+000	.1323-014	.0000	.0000	.1887-015	.5639-015	.2577-015	.3125-015
.5333+000	.2141-015	.0000	.0000	.7290-017	.4161-016	.9302-016	.7222-016
.6667+000	.6756-016	.0000	.0000	.0000	.1688-016	.2560-016	.2508-016
.8000+000	.3742-016	.0000	.0000	.0000	.1106-016	.1301-016	.1334-016

.9333+000	.1772-016	.0000	.0000	.0000	.1548-017	.4747-017	.1143-016
.1067+001	.1170-016	.0000	.0000	.0000	.8451-018	.2988-017	.7870-017
.1200+001	.2493-017	.0000	.0000	.0000	.3216-018	.1027-017	.1144-017
.1333+001	.2220-017	.0000	.0000	.0000	.4416-018	.5267-018	.1252-017
.1467+001	.1821-017	.0000	.0000	.0000	.3814-018	.2014-018	.1238-017
.1600+001	.7514-018	.0000	.0000	.0000	.3093-019	.1083-018	.6122-018
.1733+001	.3313-018	.0000	.0000	.0000	.6613-019	.7335-019	.1918-018
.1867+001	.1723-018	.0000	.0000	.0000	.5360-019	.5450-019	.6424-019
.2000+001	.1717-018	.0000	.0000	.0000	.2518-019	.6812-019	.7837-019
.2133+001	.1640-018	.0000	.0000	.0000	.1760-019	.6323-019	.8322-019
.2267+001	.1390-018	.0000	.0000	.0000	.0000	.6349-019	.7551-019
.2400+001	.1105-018	.0000	.0000	.0000	.5194-020	.3375-019	.7159-019
.2533+001	.1274-018	.0000	.0000	.0000	.1150-019	.9516-020	.1064-018
.2667+001	.1527-018	.0000	.0000	.0000	.6303-020	.7995-020	.1384-018
.2800+001	.6447-019	.0000	.0000	.0000	.0000	.4667-020	.5980-019
.2933+001	.2272-019	.0000	.0000	.0000	.2506-020	.4667-020	.1555-019
.3067+001	.2914-019	.0000	.0000	.0000	.2506-020	.1572-019	.1091-019
.3200+001	.3998-019	.0000	.0000	.0000	.8958-020	.2064-019	.1038-019
.3333+001	.2035-019	.0000	.0000	.0000	.1016-019	.8478-020	.1773-020
.3467+001	.8660-020	.0000	.0000	.0000	.2247-020	.3616-020	.2797-020
.3600+001	.3675-020	.0000	.0000	.0000	.1106-020	.5147-022	.2517-020
.3733+001	.1190-020	.0000	.0000	.0000	.0000	.0000	.1190-020
.3867+001	.1671-020	.0000	.0000	.0000	.0000	.3670-021	.1304-020
.4000+001	.2562-019	.0000	.0000	.0000	.0000	.3670-021	.2525-019
.4133+001	.2479-019	.0000	.0000	.0000	.0000	.0000	.2479-019
.4267+001	.3701-018	.0000	.0000	.0000	.0000	.1916-021	.3699-018
.4400+001	.3709-018	.0000	.0000	.0000	.0000	.1916-021	.3707-018
.4533+001	.5365-020	.0000	.0000	.0000	.2477-020	.0000	.2888-020
.4667+001	.7549-020	.0000	.0000	.0000	.2477-020	.1029-020	.4043-020
.4800+001	.3805-020	.0000	.0000	.0000	.0000	.1029-020	.2776-020
.4933+001	.1399-020	.0000	.0000	.0000	.5670-021	.0000	.8319-021
.5067+001	.5670-021	.0000	.0000	.0000	.5670-021	.0000	.0000
.5200+001	.4034-021	.0000	.0000	.0000	.0000	.3902-021	.1320-022
.5333+001	.8340-021	.0000	.0000	.0000	.0000	.3902-021	.4438-021
.5467+001	.2773-020	.0000	.0000	.0000	.0000	.1048-020	.1724-020
.5600+001	.2342-020	.0000	.0000	.0000	.0000	.1048-020	.1294-020
.5733+001	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.5867+001	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.6000+001	.8124-021	.0000	.0000	.0000	.0000	.0000	.8124-021
.6133+001	.8897-021	.0000	.0000	.0000	.0000	.0000	.8897-021
.6267+001	.8410-022	.0000	.0000	.0000	.0000	.0000	.8410-022
.6400+001	.2397-021	.0000	.0000	.0000	.0000	.0000	.2397-021
.6533+001	.4705-021	.0000	.0000	.0000	.0000	.0000	.4705-021
.6667+001	.1635-020	.0000	.0000	.0000	.0000	.0000	.1635-020
.6800+001	.1397-020	.0000	.0000	.0000	.0000	.0000	.1397-020
.6933+001	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.7067+001	.0000	.0000	.0000	.0000	.0000	.0000	.0000

```
FINITE CLOUD RADIATIVE TRANSFER MODULE


   -- RADIATIVE TRANSFER THROUGH FINITE CLOUD --

   (XC,YC,ZC) = (    .0000,    .0000,    .1000) KILOMETERS
   (AE,BE,CE) = (    .1000,    .2000,    .1000)
   (XR,YR,ZR) = (    .2000,    .0000,    .0020)
   (XS,YS,ZS) = (   -.2000,    .0000,    .0020)

    INDEXP =         5          ETA     =        .950
   LAMBDA =     1.060 (MU)      KAPPA   = 4.0000+000 (KM-1)
   OMEGA 0=     1.000           TAUBAR  = 2.0000+000
   THETA0 =      80.0           PHI0    =           .0 (DEGREES)
   RHO    =      .500           LB0     =      50.000 (W/M2-SR-MU)
   TMPA   =       9.8 (DEG.C)   TMPC    =         9.8
   LD     =         0

    **THERMAL CALCULATION OF PATH RADIANCE
       BB(LAMBDA,TMPC) = 1.3071-013 W/M2-SR-MU
       BB(LAMBDA,TMPA) = 1.3071-013 W/M2-SR-MU

 PATH LENGTH   TRANSMITTANCE  PATH RADIANCE    CONTRAST
 (IN CLOUD)                   (W/M2-SR-MU)  TRANSMITTANCE
+-------------++-------------++-------------++-------------+

     .040        8.528-001      1.129-014     1.000+000
```

RUN NUMBER 3

**** EOSAEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

INDIVIDUAL MODULES SELECTED
NMMW

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	1.167	1.167
WAVELENGTH(MICRONS)	8571.428	8571.428
FREQUENCY(GHZ)	35.000	35.000

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

```
CLASS  1 = FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS  2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLASS  3 = FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLASS  4 = FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS  5 = DUST WITH VIS LT 3 KM.
CLASS  6 = DUST WITH VIS GE 3 KM.
CLASS  7 = DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM.
CLASS  8 = DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM.
CLASS  9 = DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM.
CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM.
CLASS 11 = SNOW WITH VIS LT 1 KM.
CLASS 12 = SNOW WITH 1 LE VIS LT 3 KM.
CLASS 13 = SNOW WITH 3 LE VIS LT 7 KM.
CLASS 14 = SNOW WITH VIS GE 7 KM.
CLASS 15 = NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 16 = NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 17 = VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 18 = VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLASS 19 = CEILING HEIGHT LT 300 M.
CLASS 20 = CEILING HEIGHT LT 1000 M.
CLASS 21 = NO CEILING.
CLASS 22 = ALL CONDITIONS COMBINED.
```

EOSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS DURING SPRING AT 03-09 (LST).

CLASS NO.	FREQCY CLASS (%)	MEAN TEMP (C)	MEAN DP (C)	MEAN AH (GM/CU.M)	MEAN RH (%)	MEAN VIS (KM)	MEAN PRESS (MB)	MEAN/STDEV WNDVEL (MPS)	MEAN CLDHT (KM)	MEAN/STDEV CLDCVR (%)	FREQCY A (%)	FREQCY B (%)	FREQCY C (%)	FREQCY D (%)	FREQCY E (%)	FREQCY F (%)
1	4.5	3.5	3.2	6.3	98.2	.365	1016.7	1.9/ 1.7	.929	74.4/ 39.2	.0	2.6	12.9	69.2	1.4	14.0
2	8.2	4.1	3.2	6.4	93.8	1.945	1016.1	2.3/ 1.7	1.818	60.0/ 40.4	.0	4.1	19.0	52.6	5.1	19.2
3	17.9	5.1	3.3	6.4	89.0	4.502	1016.4	2.8/ 2.1	2.124	57.8/ 39.4	.0	4.3	15.4	52.4	8.9	19.0
4	9.6	5.8	3.3	6.3	84.5	9.224	1016.0	3.4/ 2.2	1.911	56.2/ 38.6	.0	3.1	12.5	53.1	11.5	19.7
5	.0	.0	.0	.0	.0	.000	.0	.0/ .0	.000	.0/ .0	.0	.0	.0	.0	.0	.0
6	.0	.8	-1.4	5.5	85.5	10.617	1012.3	4.2/ 3.7	1.279	64.0/ 38.2	.0	.0	5.9	64.7	.0	29.4
7	.6	5.9	5.5	7.3	97.6	.460	1013.4	2.9/ 2.6	.204	98.5/ 10.3	.0	.0	.3	97.8	.3	1.7
8	2.7	6.6	5.8	7.5	95.2	1.955	1011.8	3.6/ 2.3	.565	98.4/ 8.7	.0	.1	1.9	95.4	1.6	1.0
9	5.8	6.8	5.6	7.3	92.4	4.590	1010.3	4.5/ 2.7	.872	98.0/ 7.3	.0	.2	2.2	94.5	1.7	1.5
10	7.1	6.9	4.9	7.0	88.0	12.633	1008.8	5.6/ 3.0	1.340	92.7/ 14.5	.0	.1	1.8	92.8	3.5	1.8
11	.2	-1.0	-1.6	4.4	95.9	.522	1009.9	4.3/ 3.2	.428	96.9/ 13.5	.0	.9	2.7	95.5	.0	.9
12	.7	-1.1	-1.8	4.4	95.0	1.870	1009.9	4.4/ 2.7	.270	97.4/ 10.7	.0	.2	2.2	96.7	.2	.7
13	.8	-1.4	-2.7	4.1	91.2	4.436	1013.4	4.6/ 2.7	.824	94.1/ 14.4	.0	.0	2.7	93.0	2.2	2.1
14	.8	-.5	-2.5	4.2	86.9	13.761	1010.9	5.6/ 3.7	1.359	83.8/ 22.9	.0	.0	1.7	91.1	4.5	2.6
15	40.3	5.2	2.3	5.9	82.3	14.756	1015.9	3.9/ 2.6	2.293	51.9/ 37.9	.0	2.1	10.1	54.8	13.4	19.5
16	.8	14.7	12.5	11.1	87.3	12.319	1011.7	3.2/ 1.9	2.938	66.5/ 29.8	.0	6.9	17.7	47.0	12.8	15.7
17	3.6	4.0	3.8	6.6	98.4	.343	1015.5	2.4/ 1.9	.088	99.0/ 4.9	.0	.0	1.8	97.1	.4	.7
18	10.0	4.6	4.0	6.7	96.4	1.312	1014.2	3.1/ 2.2	.189	98.0/ 6.3	.0	.0	2.6	94.9	1.2	1.4
19	14.2	4.9	4.2	6.7	95.3	3.326	1013.7	3.9/ 2.6	.143	98.1/ 6.0	.0	.0	1.2	97.0	1.0	.7
20	35.4	5.2	3.8	6.5	90.9	7.312	1012.9	4.4/ 2.8	.390	94.9/ 9.4	.0	.0	2.0	93.8	2.7	1.6
21	37.2	4.2	1.8	5.8	85.1	10.680	1018.1	2.9/ 2.3	1.878	16.4/ 17.2	.0	5.6	17.9	28.5	13.4	34.6
22	100.0	5.3	3.2	6.3	87.1	9.302	1015.0	3.6/ 2.6	1.885	63.2/ 38.7	.0	2.5	10.7	61.9	9.1	15.8

CLASS NO.	FREQCY CLASS (%)	FREQCY WNDDIR 015 (%)	FREQCY WNDDIR 045 (%)	FREQCY WNDDIR 075 (%)	FREQCY WNDDIR 105 (%)	FREQCY WNDDIR 135 (%)	FREQCY WNDDIR 165 (%)	FREQCY WNDDIR 195 (%)	FREQCY WNDDIR 225 (%)	FREQCY WNDDIR 255 (%)	FREQCY WNDDIR 285 (%)	FREQCY WNDDIR 315 (%)	FREQCY WNDDIR 345 (%)	FREQCY WNDDIR VBL (%)
1	4.5	8.6	11.1	8.3	4.0	2.8	4.6	5.2	6.9	8.2	7.0	4.7	7.2	21.3
2	8.2	8.5	12.0	10.0	5.9	4.6	5.7	5.7	8.7	8.5	5.8	3.9	5.9	14.8
3	17.9	7.6	10.1	11.4	8.3	6.7	7.0	7.7	10.1	8.4	5.3	3.6	5.3	8.5
4	9.6	6.3	7.7	9.8	7.8	7.4	8.6	9.5	11.3	11.1	6.8	3.9	4.7	5.1
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	23.5	11.8	.0	5.9	5.9	11.8	11.8	5.9	5.9	5.9	11.8
7	.6	8.9	11.1	10.0	3.9	1.4	2.5	4.4	12.2	12.2	10.6	7.2	9.4	6.1
8	2.7	6.5	7.8	6.9	4.2	3.3	4.9	8.3	15.6	14.5	9.7	6.8	5.3	6.0
9	5.8	5.3	6.6	4.7	3.6	4.0	6.8	11.3	19.5	14.9	9.5	6.1	5.1	2.5
10	7.1	4.2	4.1	2.6	2.3	3.6	7.3	12.3	21.7	16.9	11.5	7.4	5.0	1.1
11	.2	10.0	12.7	3.6	1.8	1.8	1.8	3.6	3.6	13.6	12.7	11.8	10.9	11.8
12	.7	11.1	12.9	7.3	3.8	3.1	2.9	4.5	8.2	13.8	10.7	7.8	12.0	1.8
13	.8	10.1	11.5	9.5	3.5	1.9	2.7	4.1	9.3	15.4	10.1	7.8	10.7	3.5
14	.8	7.0	6.8	7.4	3.2	.6	1.9	1.9	8.5	21.9	17.5	12.1	8.7	2.6
15	40.3	7.5	9.1	9.7	6.8	4.7	6.8	8.4	13.2	10.5	7.8	5.0	5.9	4.8
16	.8	3.0	4.5	4.3	10.2	10.8	20.7	11.4	14.0	7.3	2.8	2.8	1.8	6.3
17	3.6	10.0	12.9	8.2	3.8	2.5	3.3	5.0	8.8	10.0	8.8	6.1	8.6	12.1
18	10.0	9.3	12.0	8.0	3.6	2.5	3.5	5.4	11.2	12.6	9.6	6.4	7.6	8.3
19	14.2	9.2	10.9	7.5	2.9	1.7	2.7	5.4	12.7	14.8	10.7	7.9	8.3	5.3
20	35.4	7.5	8.2	6.2	2.9	2.1	4.0	7.6	16.1	16.1	11.6	7.3	7.2	3.2
21	37.2	8.1	10.8	12.7	8.7	6.3	7.2	7.1	8.4	6.9	5.0	3.6	5.2	9.9
22	100.0	7.2	8.9	9.0	6.3	5.0	6.8	8.3	12.7	10.9	7.6	4.9	5.7	6.6

VISIBILITY
9.30 KM

```
NEAR MILLIMETER WAVE MODULE


TEMPERATURE                 5.300 DEGREES C
PRESSURE                 1015.000 MB
ABSOLUTE HUMIDITY           6.300 G/M**3
FOG DENSITY                  .500 G/M**3
RAIN RATE                   5.000 MM/HR
SNOW RATE                    .000 MM/HR
FREQUENCY                  35.000 GHZ
PATH LENGTH                  .400 KM

GAS ABSORPTION          .6838-001 DB/KM
FOG EXTINCTION          .4565+000 DB/KM
RAIN EXTINCTION         .1352+001 DB/KM
SNOW EXTINCTION         .0000     DB/KM
TRANSMISSION            .8413+000

FOG BACKSCATTER         .5724-010 M**2/M**3
RAIN BACKSCATTER        .1026-003 M**2/M**3
SNOW BACKSCATTER        .0000     M**2/M**3
TOTAL BACKSCATTER       .1026-003 M**2/M**3


TEMPERATURE                 5.300 DEGREES C
PRESSURE                 1015.000 MB
ABSOLUTE HUMIDITY           6.300 G/M**3
FOG DENSITY                  .500 G/M**3
RAIN RATE                    .000 MM/HR
SNOW RATE                   5.000 MM/HR
FREQUENCY                  35.000 GHZ
PATH LENGTH                  .400 KM

GAS ABSORPTION          .6838-001 DB/KM
FOG EXTINCTION          .4565+000 DB/KM
RAIN EXTINCTION         .0000     DB/KM
SNOW EXTINCTION         .4708+001 DB/KM
TRANSMISSION            .6175+000

FOG BACKSCATTER         .5724-010 M**2/M**3
RAIN BACKSCATTER        .0000     M**2/M**3
SNOW BACKSCATTER        .3277-003 M**2/M**3
TOTAL BACKSCATTER       .3277-003 M**2/M**3

COMBINED TRANSMISSION FOR THE SELECTED MODULES =  .5195+000
```

RUN NUMBER 4

**** EOSAEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

INDIVIDUAL MODULES SELECTED
SPOT
LT4M

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	2010.000	2710.000
WAVELENGTH(MICRONS)	3.690	4.975
FREQUENCY(GHZ)	60300.000	81300.000

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

```
CLASS  1 = FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS  2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLASS  3 = FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLASS  4 = FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS  5 = DUST WITH VIS LT 3 KM.
CLASS  6 = DUST WITH VIS GE 3 KM.
CLASS  7 = DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM.
CLASS  8 = DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM.
CLASS  9 = DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM.
CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM.
CLASS 11 = SNOW WITH VIS LT 1 KM.
CLASS 12 = SNOW WITH 1 LE VIS LT 3 KM.
CLASS 13 = SNOW WITH 3 LE VIS LT 7 KM.
CLASS 14 = SNOW WITH VIS GE 7 KM.
CLASS 15 = NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 16 = NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 17 = VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 18 = VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLASS 19 = CEILING HEIGHT LT 300 M.
CLASS 20 = CEILING HEIGHT LT 1000 M.
CLASS 21 = NO CEILING.
CLASS 22 = ALL CONDITIONS COMBINED.
```

EOSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS DURING SPRING AT 03-09 (LST).

CLASS NO.	FREQCY CLASS (%)	MEAN TEMP (C)	MEAN DP (C)	MEAN AH (GM/CU.M)	MEAN RH (%)	MEAN VIS (KM)	MEAN PRESS (NB)	MEAN/STDEV WNDVEL (MPS)	MEAN CLDHT (KM)	MEAN/STDEV CLDCVR (%)	FREQCY A (%)	FREQCY B (%)	FREQCY C (%)	FREQCY D (%)	FREQCY E (%)	FREQCY F (%)
1	4.5	3.5	3.2	6.3	98.2	.365	1016.7	1.9/ 1.7	.929	74.4/ 39.2	.0	2.6	12.9	69.2	1.4	14.0
2	8.2	4.1	3.2	6.4	93.8	1.945	1016.1	2.3/ 1.7	1.818	60.0/ 40.4	.0	4.1	19.0	52.6	5.1	19.2
3	17.9	5.1	3.3	6.4	89.0	4.502	1016.4	2.8/ 2.1	2.124	57.8/ 39.4	.0	4.3	15.4	52.4	8.9	19.0
4	9.6	5.8	3.3	6.3	84.5	9.224	1016.0	3.4/ 2.2	1.911	56.2/ 38.6	.0	3.1	12.5	53.1	11.5	19.7
5	.0	.0	.0	.0	.0	.000	.0	.0/ .0	.000	.0/ .0	.0	.0	.0	.0	.0	.0
6	.0	.8	-1.4	5.5	85.5	10.617	1012.3	4.2/ 3.7	1.279	64.0/ 38.2	.0	.0	5.9	64.7	.0	29.4
7	.6	5.9	5.5	7.3	97.6	.460	1013.4	2.9/ 2.6	.204	98.5/ 10.3	.0	.0	.3	97.8	.3	1.7
8	2.7	6.6	5.8	7.5	95.2	1.955	1011.8	3.6/ 2.3	.565	98.4/ 8.7	.0	.1	1.9	95.4	1.6	1.0
9	5.8	6.8	5.6	7.3	92.4	4.590	1010.3	4.5/ 2.7	.872	98.0/ 7.3	.0	.2	2.2	94.5	1.7	1.5
10	7.1	6.9	4.9	7.0	88.0	12.633	1008.8	5.6/ 3.0	1.340	92.7/ 14.5	.0	.1	1.8	92.8	3.5	1.8
11	.2	-1.0	-1.6	4.4	95.9	.522	1009.9	4.3/ 3.2	.428	96.9/ 13.5	.0	.9	2.7	95.5	.0	.9
12	.7	-1.1	-1.8	4.4	95.0	1.870	1009.9	4.4/ 2.7	.270	97.4/ 10.7	.0	.2	2.2	96.7	.2	.7
13	.8	-1.4	-2.7	4.1	91.2	4.436	1013.4	4.6/ 2.7	.824	94.1/ 14.4	.0	.0	2.7	93.0	2.3	2.1
14	.8	-.5	-2.5	4.2	86.9	13.761	1010.9	5.6/ 3.7	1.359	83.8/ 22.9	.0	.0	1.7	91.1	4.5	2.6
15	40.3	5.2	2.3	5.9	82.3	14.756	1015.9	3.9/ 2.6	2.293	51.9/ 37.9	.0	2.1	10.1	54.8	13.4	19.5
16	.8	14.7	12.5	11.1	87.3	12.319	1011.7	3.2/ 1.9	2.938	66.5/ 29.8	.0	6.9	17.7	47.0	12.8	15.7
17	3.6	4.0	3.8	6.6	98.4	.343	1015.5	2.4/ 1.9	.088	99.0/ 4.9	.0	.0	1.8	97.1	.4	.7
18	10.0	4.6	4.0	6.7	96.4	1.312	1014.2	3.1/ 2.2	.189	98.0/ 6.3	.0	.0	2.6	94.9	1.2	1.4
19	14.2	4.9	4.2	6.7	95.3	3.326	1013.7	3.9/ 2.6	.143	98.1/ 6.0	.0	.0	1.2	97.0	1.0	.7
20	35.4	5.2	3.8	6.5	90.9	7.312	1012.9	4.4/ 2.8	.390	94.9/ 9.4	.0	.0	2.0	93.8	2.7	1.6
21	37.2	4.2	1.8	5.8	85.1	10.680	1018.1	2.9/ 2.3	1.878	16.4/ 17.2	.0	5.6	17.9	28.5	13.4	34.6
22	100.0	5.3	3.2	6.3	87.1	9.302	1015.0	3.6/ 2.6	1.885	63.2/ 38.7	.0	2.5	10.7	61.9	9.1	15.8

CLASS NO.	FREQCY CLASS (%)	FREQCY WNDDIR 015 (%)	FREQCY WNDDIR 045 (%)	FREQCY WNDDIR 075 (%)	FREQCY WNDDIR 105 (%)	FREQCY WNDDIR 135 (%)	FREQCY WNDDIR 165 (%)	FREQCY WNDDIR 195 (%)	FREQCY WNDDIR 225 (%)	FREQCY WNDDIR 255 (%)	FREQCY WNDDIR 285 (%)	FREQCY WNDDIR 315 (%)	FREQCY WNDDIR 345 (%)	FREQCY WNDDIR VBL (%)
1	4.5	8.6	11.1	8.3	4.0	2.8	4.6	5.2	6.9	8.2	7.0	4.7	7.2	21.3
2	8.2	8.5	12.0	10.0	5.9	4.6	5.7	5.7	8.7	8.5	5.8	3.9	5.9	14.8
3	17.9	7.6	10.1	11.4	8.3	6.7	7.0	7.7	10.1	8.4	5.3	3.6	5.3	8.5
4	9.6	6.3	7.7	9.8	7.8	7.4	8.6	9.5	11.3	11.1	6.8	3.9	4.7	5.1
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	23.5	11.8	.0	5.9	5.9	11.8	11.8	5.9	5.9	5.9	11.8
7	.6	8.9	11.1	10.0	3.9	1.4	2.5	4.4	12.2	12.2	10.6	7.2	9.4	6.1
8	2.7	6.5	7.8	6.9	4.2	3.3	4.9	8.3	15.6	14.5	9.7	6.8	5.3	6.0
9	5.8	5.3	6.6	4.7	3.6	4.0	6.8	11.3	19.5	14.9	9.5	6.1	5.1	2.5
10	7.1	4.2	4.1	2.6	2.3	3.6	7.3	12.3	21.7	16.9	11.5	7.4	5.0	1.1
11	.2	10.0	12.7	3.6	1.8	1.8	1.8	3.6	3.6	13.6	12.7	11.8	10.9	11.8
12	.7	11.1	12.9	7.3	3.8	3.1	2.9	4.5	8.2	13.8	10.7	7.8	12.0	1.8
13	.8	10.1	11.5	9.5	3.5	1.9	2.7	4.1	9.3	15.4	10.1	7.8	10.7	3.5
14	.8	7.0	6.8	7.4	3.2	.6	1.9	1.9	8.5	21.9	17.5	12.1	8.7	2.6
15	40.3	7.5	9.1	9.7	6.8	4.7	6.8	8.4	13.2	10.5	7.8	5.0	5.9	4.8
16	.8	3.0	4.5	4.3	10.2	10.8	20.7	11.4	14.0	7.3	2.8	2.8	1.8	6.3
17	3.6	10.0	12.9	8.2	3.8	2.5	3.3	5.0	8.8	10.0	8.8	6.1	8.6	12.1
18	10.0	9.3	12.0	8.0	3.6	2.5	3.5	5.4	11.2	12.6	9.6	6.4	7.6	8.3
19	14.2	9.2	10.9	7.5	2.9	1.7	2.7	5.4	12.7	14.8	10.7	7.9	8.3	5.3
20	35.4	7.5	8.2	6.2	2.9	2.1	4.0	7.6	16.1	16.1	11.6	7.3	7.2	3.2
21	37.2	8.1	10.8	12.7	8.7	6.3	7.2	7.1	8.4	6.9	5.0	3.6	5.2	9.9
22	100.0	7.2	8.9	9.0	6.3	5.0	6.8	8.3	12.7	10.9	7.6	4.9	5.7	6.6

VISIBILITY
9.30 KM

SPOT CONTRAST MODULE

```
SPOT CONTROL CARDS READ FOR THIS RUN:

ENVR        .3000+001 .2000+001 .2000+001 .4000+001 .1000+001 .1000+001
EMIS        .1000+001 .2830+003 .9500+000 .2950+003 .0000     .0000
ATM         .6500+002 .0000     .0000     .0000     .0000     .0000
TARG        .4000+000 .4500+002 .9000+002 .4500+002 .0000     .0000
REFL        .5000+000 .5000+000 .0000     .5000-001 .0000     .0000
SENS        .2000-002 .9000+002 .2700+003 .1000+001 .0000     .0000
GO          .0000     .0000     .0000     .0000     .0000     .0000
DIVISION LIMITS CHANGED FROM     40.000 TO     60.000
                    INPUT RESPONSE FUNCTION
              WAVELENGTH     R FUNCTION
               .3500+001  .7800+000
               .3600+001  .8300+000
               .3700+001  .8700+000
               .3800+001  .9200+000
               .3900+001  .9600+000
               .4000+001  .9800+000
               .4100+001  .9700+000
               .4300+001  .9600+000
               .4500+001  .9500+000
               .4700+001  .9400+000
               .4900+001  .9300+000
               .5000+001  .9300+000
```

```
*****************************************************************************
*                                                                           *
*                     SPOT DIAGNOSTIC MESSAGES FOLLOW:                      *
*                                                                           *
*   1.  NO DIRECT  SUNLIGHT  INCIDENT WITHIN RECEIVER'S FIELD OF VIEW.      *
*                                                                           *
*****************************************************************************
```

DEFINITION OF CONTROL PARAMETERS FOLLOWS:

PARAMETER	VALUE	DESCRIPTION
ISORC	3	SUNLIGHT AND EMISSION
ITARG	2	TARGET REFLECTANCE / EMISSION
MODEL	4	SUBARCTIC SUMMER
IHAZE	2	

CLOUD BOTTOM HEIGHT = .000 KM

SOURCE INTENSITIES

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	SUNLIGHT SOURCE STRENGTH (WATTS M-2 MICRON-1)	TARGET SOURCE STRENGTH (WATTS M-2 MICRON-1 SR-1)	GROUND SOURCE STRENGTH (WATTS M-2 MICRON-1 SR-1)
4.9751+000	2010	3.8787+000	2.0519+000	.0000
4.8309+000	2070	4.3600+000	1.7739+000	.0000
4.6948+000	2130	4.8846+000	1.5271+000	.0000
4.5662+000	2190	5.4551+000	1.3095+000	.0000
4.4444+000	2250	6.0741+000	1.1187+000	.0000
4.3290+000	2310	6.7442+000	9.5228-001	.0000
4.2194+000	2370	7.4682+000	8.0789-001	.0000
4.1152+000	2430	8.2488+000	6.8321-001	.0000
4.0161+000	2490	9.0890+000	5.7601-001	.0000
3.9216+000	2550	9.9916+000	4.8422-001	.0000
3.8314+000	2610	1.0960+001	4.0593-001	.0000
3.7453+000	2670	1.1996+001	3.3940-001	.0000

COMPONENTS FOR RADIANCE FROM TARGET

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET EMISSION	TARGET REFLECTANCE	PARTIAL ATMOSPHERIC EMISSION	PARTIAL PATH RADIANCE	TOTAL TARGET RADIANCE
4.9751+000	2010	8.3677-001	3.5950-003	6.9555-001	2.6453-004	1.5362+000
4.8309+000	2070	7.0848-001	1.4220-003	6.1320-001	1.0494-004	1.3232+000
4.6948+000	2130	7.8964-001	7.6302-003	3.2337-001	4.8794-004	1.1211+000
4.5662+000	2190	6.8010-001	8.1827-003	2.7173-001	4.8895-004	9.6050-001
4.4444+000	2250	2.2254-001	6.0905-006	6.2320-001	4.6230-007	8.4575-001
4.3290+000	2310	6.5140-004	1.6137-015	7.3146-001	1.0483-024	7.3211-001
4.2194+000	2370	8.7626-003	2.4889-011	6.0666-001	3.2369-016	6.1542-001
4.1152+000	2430	4.3506-001	3.1147-002	5.1388-002	1.2642-003	5.1886-001
4.0161+000	2490	3.7393-001	4.9777-002	3.5402-002	1.9074-003	4.6102-001
3.9216+000	2550	3.1054-001	6.7674-002	3.3520-002	2.6148-003	4.1435-001
3.8314+000	2610	2.6043-001	8.2765-002	2.7776-002	3.2195-003	3.7419-001
3.7453+000	2670	2.1124-001	7.8440-002	2.9776-002	3.1049-003	3.2256-001

COMPONENTS FOR BACKGROUND RADIANCE

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	GROUND EMISSION	GROUND REFLECTANCE	TOTAL ATMOSPHERIC EMISSION	TOTAL PATH RADIANCE	TOTAL BACKGROUND RADIANCE
4.9751+000	2010	.0000	.0000	1.6433+000	1.3950-003	1.6447+000
4.8309+000	2070	.0000	.0000	1.4091+000	3.3501-004	1.4094+000
4.6948+000	2130	.0000	.0000	1.2032+000	2.6254-003	1.2059+000
4.5662+000	2190	.0000	.0000	1.0234+000	4.0099-003	1.0274+000
4.4444+000	2250	.0000	.0000	8.6716-001	1.0520-006	8.6716-001
4.3290+000	2310	.0000	.0000	7.3217-001	1.0655-024	7.3217-001
4.2194+000	2370	.0000	.0000	6.1611-001	3.5637-016	6.1611-001
4.1152+000	2430	.0000	.0000	5.1679-001	1.3371-002	5.3016-001
4.0161+000	2490	.0000	.0000	4.3216-001	2.4559-002	4.5672-001
3.9216+000	2550	.0000	.0000	3.6035-001	3.6939-002	3.9728-001
3.8314+000	2610	.0000	.0000	2.9963-001	4.6921-002	3.4655-001
3.7453+000	2670	.0000	.0000	2.4849-001	4.1007-002	2.8950-001

DIRECT SUNLIGHT

(WATTS M-2 MICRON-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	SUNLIGHT SOURCE STRENGTH	SUNLIGHT FLUX
4.9751+000	2010	3.8787+000	.0000
4.8309+000	2070	4.3600+000	.0000
4.6948+000	2130	4.8846+000	.0000
4.5662+000	2190	5.4551+000	.0000
4.4444+000	2250	6.0741+000	.0000
4.3290+000	2310	6.7442+000	.0000
4.2194+000	2370	7.4682+000	.0000
4.1152+000	2430	8.2488+000	.0000
4.0161+000	2490	9.0890+000	.0000
3.9216+000	2550	9.9916+000	.0000
3.8314+000	2610	1.0960+001	.0000
3.7453+000	2670	1.1996+001	.0000

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

6 6

END
DATE
FILMED
6 82
DTIC

TOTAL RADIANCE

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET	BACKGROUND	CONTRAST RATIO
4.9751+000	2010	1.5362+000	1.6447+000	-6.5959-002
4.8309+000	2070	1.3232+000	1.4094+000	-6.1179-002
4.6948+000	2130	1.1211+000	1.2059+000	-7.0270-002
4.5662+000	2190	9.6050-001	1.0274+000	-6.5111-002
4.4444+000	2250	8.4575-001	8.6716-001	-2.4691-002
4.3290+000	2310	7.3211-001	7.3217-001	-7.7684-005
4.2194+000	2370	6.1542-001	6.1611-001	-1.1165-003
4.1152+000	2430	5.1886-001	5.3016-001	-2.1316-002
4.0161+000	2490	4.6102-001	4.5672-001	9.4083-003
3.9216+000	2550	4.1435-001	3.9728-001	4.2959-002
3.8314+000	2610	3.7419-001	3.4655-001	7.9745-002
3.7453+000	2670	3.2256-001	2.8950-001	1.1420-001

DETECTOR-RESPONSE WAVELENGTH-INTEGRATED

(WATTS M-2 SR-1)

TARGET EMISSION	4.9117-002
TARGET REFLECTANCE	1.8157-003
PARTIAL ATMOSPHERIC EMISSION	4.3029-002
PARTIAL PATH RADIANCE	7.7395-005
TOTAL TARGET RADIANCE	9.4039-002
GROUND EMISSION	.0000
GROUND REFLECTANCE	.0000
TOTAL ATMOSPHERIC EMISSION	9.7352-002
TOTAL PATH RADIANCE	9.1328-004
TOTAL BACKGROUND RADIANCE	9.8265-002

****

CONTRAST -4.3004-002

****

DIRECT SUNLIGHT .0000
(WATTS M-2)

LT4M ATMOSPHERIC TRANSMISSION MODULE

PROGRAM WILL BE EXECUTED IN THE TRANSMISSION MODE

HORIZONTAL PATH, ALTITUDE = .002 KM,RANGE = .400 KM

MODEL ATMOSPHERE 4 = SUB-ARCTIC SUMMER

FREQUENCY RANGE V1= 2010.0 CM-1 TO V2= 2710.0 CM-1 FOR DV = 60.0 CM-1 (3.69 - 4.98 MICRONS)

EQUIVALENT SEA LEVEL ABSORBER AMOUNTS

	WATER VAPOUR GM CM-2	CO2 ETC. KM	OZONE ATM CM	NITROGEN (CONT) KM	H2O (CONT) GM CM-2	MOL SCAT KM	OZONE(U-V) ATM CM
W(1-6,8)=	.355+000	.372+000	.905-003	.295+000	.595-002	.380+000	.915-003
W(10)=					.546-001		

	NITRIC ACID	SO2	NH3	NO2		
W(11-16)=	.000	.834-002	.779-002	.337-002	.433-002	.707-004

FREQ CM-1	WAVELENGTH MICRONS	H2O TRANS	CO2+ TRANS	OZONE TRANS	N2 C TRANS	H2O C TRANS	MOL S TRANS	NITRIC TRANS	SO2 TRANS	HNO3 TRANS	NO2 TRANS	INTEGRATED ABSORPTION	TOTAL TRANS	AEROSOL TRANS
2010	4.9751	.6090	.9987	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	12.6982	.5767	.94820
2070	4.8309	.7608	.9529	.9996	1.0000	.8220	1.0000	1.0000	1.0000	1.0000	1.0000	38.8084	.5648	.94820
2130	4.6948	.8649	.9853	.9990	.9990	.9066	1.0000	1.0000	1.0000	1.0000	1.0000	54.9334	.7313	.94820
2190	4.5662	.9398	.8313	1.0000	.9915	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	70.8646	.7345	.94820
2250	4.4444	.9367	.3230	1.0000	.9807	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	113.9851	.2813	.94820
2310	4.3290	.9769	.0011	1.0000	.9652	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	173.9270	.0010	.94820
2370	4.2194	.9925	.0172	1.0000	.9675	.9791	1.0000	1.0000	1.0000	1.0000	1.0000	233.0067	.0153	.94820
2430	4.1152	.9977	.9941	1.0000	.9723	.9848	1.0000	1.0000	1.0000	1.0000	1.0000	238.9729	.9006	.94820
2490	4.0161	.9986	.9941	1.0000	.9890	.9890	1.0000	1.0000	.9971	1.0000	1.0000	243.8880	.9181	.94820
2550	3.9216	.9903	.9768	1.0000	.9972	.9916	1.0000	1.0000	1.0000	1.0000	1.0000	249.4692	.9070	.94820
2610	3.8314	.9663	.9985	1.0000	.9993	.9925	1.0000	1.0000	1.0000	1.0000	1.0000	255.0312	.9073	.94820
2670	3.7453	.9383	.9981	1.0000	.9996	.9916	1.0000	1.0000	1.0000	1.0000	1.0000	262.2208	.8802	.94820

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

INTEGRATED ASORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358

```
                              LT4M ATMOSPHERIC TRANSMISSION MODULE
PROGRAM WILL BE EXECUTED IN THE EMISSION MODE

      HORIZONTAL PATH, ALTITUDE =    .002 KM,RANGE =    .400 KM
            MODEL ATMOSPHERE 4 = SUB-ARCTIC SUMMER
      FREQUENCY RANGE V1=  2010.0 CM-1 TO V2=  2710.0 CM-1 FOR DV =  60.0 CM-1   (  3.69 -  4.98 MICRONS )
      EQUIVALENT SEA LEVEL ABSORBER AMOUNTS
            WATER VAPOUR     CO2 ETC.     OZONE    NITROGEN (CONT)  H2O (CONT)     MOL  SCAT     OZONE(U-V)
              GM CM-2          KM        ATM CM          KM          GM CM-2        KM           ATM CM
      W(1-6,8)=   .355+000     .372+000     .905-003     .295+000     .595-002     .380+000     .915-003
       W(10)=                                                         .546-001

             NITRIC ACID      SO2          NH3          NO2

      W(11-16)=   .000         .834-002     .779-002     .337-002     .433-002     .707-004

            CUMULATIVE ABSORBER AMOUNTS FOR THE ATMOSPHERIC PATH
     H2O      CO2+      O3        N2       H2O C     MOL S     O3 UV     H2O C     HNO3       SO2       NH3       NO2      TAVE
 1 .355+000 .372+000 .905-003 .295+000 .595-002 .380+000 .915-003 .546-001 .000      .834-002 .779-002 .337-002  287.000
```

FR(CM-1)	WVL(MICRON)	RADIANCE(WATTS/CM2-STER-XXX) PER CM-1	RADIANCE(WATTS/CM2-STER-XXX) PER MICRON	INTEGRAL	TRANS	---- AERO TRAN ---- EXTN	---- AERO TRAN ---- ABS
2010.0	4.975124	.17216-006	.69555-004	.51648-005	.576727	.94820	.98942
2070.0	4.830918	.14311-006	.61320-004	.13751-004	.564830	.94820	.98942
2130.0	4.694836	.71276-007	.32337-004	.18028-004	.731250	.94820	.98942
2190.0	4.566210	.56657-007	.27173-004	.21427-004	.734479	.94820	.98942
2250.0	4.444444	.12310-006	.62320-004	.28813-004	.281325	.94820	.98942
2310.0	4.329004	.13708-006	.73146-004	.37038-004	.000967	.94820	.98942
2370.0	4.219409	.10801-006	.60666-004	.43518-004	.015339	.94820	.98942
2430.0	4.115226	.87026-008	.51388-005	.44040-004	.900563	.94820	.98942
2490.0	4.016064	.57098-008	.35402-005	.44383-004	.918083	.94820	.98942
2550.0	3.921569	.51549-008	.33520-005	.44692-004	.906979	.94820	.98942
2610.0	3.831418	.40774-008	.27776-005	.44937-004	.907301	.94820	.98942
2670.0	3.745318	.41768-008	.29776-005	.45188-004	.880173	.94820	.98942

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

RADMIN 2610.000 .40774-008
RADMAX 2010.000 .17216-006

INTEGRATED ASORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358
INTEGRATED RADIANCE = .45188-004 WATT CM -2 SR

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .4711-001

END EOSAEL RUN

ELECTRO-OPTICS DISTRIBUTION LIST

Commander
US Army Aviation School
Fort Rucker, AL 36362

Commander
US Army Aviation Center
ATTN: ATZQ-D-MA (Mr. Oliver N. Heath)
Fort Rucker, AL 36362

Commander
US Army Aviation Center
ATTN: ATZQ-D-MS (Mr. Donald Wagner)
Fort Rucker, AL 36362

NASA/Marshall Space Flight Center
ATTN: ES-83 (Otha H. Vaughan, Jr.)
Huntsville, AL 35812

NASA/Marshall Space Flight Center
Atmospheric Sciences Division
ATTN: Code ES-81 (Dr. William W. Vaughan)
Huntsville, AL 35812

Nichols Research Corporation
ATTN: Dr. Lary W. Pinkley
4040 South Memorial Parkway
Huntsville, AL 35802

John M. Hobbie
c/o Kentron International
2003 Byrd Spring Road
Huntsville, AL 35802

Mr. Ray Baker
Lockheed-Missile & Space Company
4800 Bradford Blvd
Huntsville, AL 35807

Commander
US Army Missile Command
ATTN: DRSMI-OG (Mr. Donald R. Peterson)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-OGA (Dr. Bruce W. Fowler)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REL (Dr. George Emmons)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Huey F. Anderson)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Mr. Maxwell W. Harper)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Mr. Gene Widenhofer)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RHC (Dr. Julius Q. Lilly)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
Redstone Scientific Information Center
ATTN: DRSMI-RPRD (Documents Section)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRA (Dr. Oskar Essenwanger)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRO (Mr. Charles Christensen)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRO (Dr. George A. Tanton)
Redstone Arsenal, AL 35809

Commander
US Army Communications Command
ATTN: CC-OPS-PP
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-CS (Mr. Richard G. Cundy)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-MD (Mr. Harry Wilder)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CS-C (2LT Coffman)
Fort Huachuca, AZ 85613

Commander
US Army Yuma Proving Ground
ATTN: STEYP-MSA-TL
Bldg 2105
Yuma, AZ 85364

Northrop Corporation
Electro-Mechanical Division
ATTN: Dr. Richard D. Tooley
500 East Orangethorpe Avenue
Anaheim, CA 92801

Commander
Naval Weapons Center
ATTN: Code 3918 (Dr. Alexis Shlanta)
China Lake, CA 93555

Hughes Helicopters
Army Advanced Attack Helicopter Weapons
ATTN: Mr. Charles R. Hill
Centinela and Teale Streets
Bldg 305, MS T-73A
Culter City, CA 90230

Commander
US Army Combat Developments
Experimentation Command
ATTN: ATEC-PL-M (Mr. Gary G. Love)
Fort Ord, CA 93941

SRI International
ATTN: K2060/Dr. Edward E. Uthe
333 Ravenswood Avenue
Menlo Park, CA 94025

SRI International
ATTN: Mr. J. E. Van der Laan
333 Ravenswood Avenue
Menlo Park, CA 94025

Joane May
Naval Environmental Prediction
Research Facility (NEPRF)
ATTN: Library
Monterey, CA 93940

Sylvania Systems Group,
Western Division
GTE Products Corporation
ATTN: Technical Reports Library
P.O. Box 205
Mountain View, CA 94042

Sylvania Systems Group
Western Division
GTE Products Corporation
ATTN: Mr. Lee W. Carrier
P.O. Box 188
Mountain View, CA 94042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3250-3 (R. de Violini)
Point Mugu, CA 93042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3253 (Terry E. Battalino)
Point Mugu, CA 93042

Effects Technology Inc.
ATTN: Mr. John D. Carlyle
5383 Hollister Avenue
Santa Barbara, CA 93111

Commander
Naval Ocean Systems Center
ATTN: Code 532 (Dr. Juergen Richter)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 5322 (Mr. Herbert G. Hughes)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 4473 (Tech Library)
San Diego, CA 92152

The RAND Corporation
ATTN: Ralph Huschke
1700 Main Street
Santa Monica, CA 90406

Particle Measuring Systems, Inc.
ATTN: Dr. Robert G. Knollenberg
1855 South 57th Court
Boulder, CO 80301

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: Library, R-51, Technical Reports
325 Broadway
Boulder, CO 80303

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: R45X3 (Dr. Vernon E. Derr)
Boulder, CO 80303

US Department of Commerce
National Telecommunications and
Information Administration
Institute for Telecommunication Sciences
ATTN: Code 1-3426 (Dr. Hans J. Liebe)
Boulder, CO 80303

AFATL/DLODL
Technical Library
Eglin AFB, FL 32542

Commanding Officer
Naval Training Equipment Center
ATTN: Technical Information Center
Orlando, FL 32813

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. Robert W. McMillan
Atlanta, GA 30332

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. James C. Wiltse
Atlanta, GA 30332

Commandant
US Army Infantry Center
ATTN: ATSH-CD-MS-E (Mr. Robert McKenna)
Fort Benning, GA 31805

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-CS
Fort Gordon, GA 30905

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-O
Fort Gordon, GA 30905

USAFETAC/DNE
ATTN: Mr. Charles Glauber
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DNDP (LTC Kit G. Cottrell)
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DOOF (MAJ Robert Wright)
Scott AFB, IL 62225

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-CAA-Q (Mr. H. Kent Pickett)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Robert DeKinder, Jr.)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Mr. Kent I. Johnson)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-WE (LTC Darrell Holland)
Fort Leavenworth, KS 66027

President
USAARENBD
ATTN: ATZK-AE-TA (Dr. Charles R. Leake)
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-MS
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-SD
Fort Knox, KY 40121

Aerodyne Research Inc.
ATTN: Dr. John F. Ebersole
Crosby Drive
Bedford, MA 01730

Commander
Air Force Geophysics Laboratory
ATTN: OPA (Dr. Robert W. Fenn)
Hanscom AFB, MA 01731

Commander
Air Force Geophysics Laboratory
ATTN: OPI (Dr. Robert A. McClatchey)
Hanscom AFB, MA 01731

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. T. J. Goblick, B-370
P.O. Box 73
Lexington, MA 02173

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. Michael Gruber
P.O. Box 73
Lexington, MA 02173

Raytheon Company
Equipment Division
ATTN: Dr. Charles M. Sonnenschein
430 Boston Post Road
Wayland, MA 01778

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-BLB (Mr. Richard McGee)
Aberdeen Proving Ground, MD 21005

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Dr. Edward Stuebing)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Mr. Joseph Vervier)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLY-A (Mr. Ronnald Pennsyle)
Aberdeen Proving Ground, MD 21010

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-TSB-S (STINFO)
Aberdeen Proving Ground, MD 21005

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CCM (W. H. Pepper)
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CG/DRDEL-DC/DRDEL-CS
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CT
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-PAO (M. Singleton)
2800 Powder Mill Road
Adelphi, MD 20783

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK
(Dr. Anthony Van de Wal, Jr.)
Aberdeen Proving Ground, MD 21005

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK-T (Mr. Sidney Gerard)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSTE-AD-M (Mr. Warren M. Baity)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSTE-AD-M (Dr. Norman E. Pentz)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-AAM (Mr. William Smith)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-CS (Mr. Philip H. Beavers)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GB (Wilbur L. Warfield)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GP (Mr. Fred Campbell)
Aberdeen Proving Ground, MD 21005

Director
US Army Material Systems Analysis Activity
ATTN: DRXSY-GP (H. Stamper)
Aberdeen Proving Grounds, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GS
(Mr. Michael Starks/Mr. Julian Chernick)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-J (Mr James F. O'Bryon)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-LM (Mr. Robert M. Marchetti)
Aberdeen Proving Ground, MD 21005

Commander
Harry Diamond Laboratories
ATTN: Dr. William W. Carter
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM (Mr. Robert McCoskey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Robert Humphrey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Z. G. Sztankay)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Joseph Nemarich)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Air Force Systems Command
ATTN: WER (Mr. Richard F. Picanso)
Andrews AFB, MD 20334

Martin Marietta Laboratories
ATTN: Jar Mo Chen
1450 South Rolling Road
Baltimore, MD 21227

Commander
US Army Concepts Analysis Agency
ATTN: CSCA-SMC (Mr. Hal E. Hock)
8120 Woodmont Avenue
Bethesda, MD 20014

Director
National Security Agency
ATTN: R52/Dr. Douglas Woods
Fort George G. Meade, MD 20755

Chief
Intelligence Materiel Development
& Support Office
US Army Electronic Warfare Laboratory
ATTN: DELEW-I (LTC Kenneth E. Thomas)
Fort George G. Meade, MD 20755

The Johns Hopkins University
Applied Physics Laboratory
ATTN: Dr. Michael J. Lun
John Hopkins Road
Laurell, MD 20810

Dr. Stephen T. Hanley
1720 Rhodesia Avenue
Oxon Hill, MD 20022

Science Applications Inc.
ATTN: Mr. G. D. Currie
15 Research Drive
Ann Arbor, MI 48103

Science Applications Inc.
ATTN: Dr. Robert E. Turner
15 Research Drive
Ann Arbor, MI 48103

Commander
US Army Tank-Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Harry Young)
Warren, MI 48090

Commander
US Army Tank Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Wallace Mick, Jr.)
Warren, MI 48090

Dr. A. D. Belmont
Research Division
Control Data Corporation
P.O. Box 1249
Minneapolis, MN 55440

Director
US Army Engr Waterways Experiment Station
ATTN: WESEN (Mr. James Mason)
P.O. Box 631
Vicksburg, MS 39180

Dr. Jerry Davis
Department of Marine, Earth
and Atmospheric Sciences
North Carolina State University
Raleigh, NC 27650

Commander
US Army Research Office
ATTN: DRXRO-GS (Dr. Leo Alpert)
P.O. Box 12211
Research Triangle Park, NC 27709

Commander
US Army Research Office
ATTN: DRXRO-PP (Brenda Mann)
P.O. Box 12211
Research Triangle Park, NC 27709

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RD (Dr. K. F. Sterrett)
Hanover, NH 03755

Commander/Director
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. George Aitken)
Hanover, NH 03755

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. Roger H. Berger)
Hanover, NH 03755

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-AC (Mr. James Greenfield)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-TSS (Bldg #59)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. Peteris Jansons)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. G. H. Waldron)
Dover, NJ 07801

Deputy Joint Project Manager
for Navy/USMC SAL GP
ATTN: DRCPM-CAWS-NV (CPT Joseph Miceli)
Dover, NJ 07801

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-I (Mr. David Longinotti)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-PE (Mr. Ben A. Di Campli)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-R-S (Mr. Donald L. Foiani)
Fort Monmouth, NJ 07703

Director
US Army Electronics Technology &
Devices Laboratory
ATTN: DELET-DD (S. Danko)
Fort Monmouth, NJ 07703

Project Manager
FIREFINDER/REMBASS
ATTN: DRCPM-FFR-TM (Mr. John M. Bialo)
Fort Monmouth, NJ 07703

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-SA (Dr. Walter S. McAfee)
Fort Monmouth, NJ 07703

OLA, 2WS (MAC)
Holloman AFB, NM 88330

Commander
Air Force Weapons Laboratory
ATTN: AFWL/WE (MAJ John R. Elrick)
Kirtland, AFB, NM 87117

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL (Dolores Anguiano)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. Louie Dominguez)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. William J. Leach)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TGP (Mr. Roger F. Willis)
White Sands Missile Range, NM 88002

Director
Office of Missile Electronic Warfare
ATTN: DELEW-M-STO (Dr. Steven Kovel)
White Sands Missile Range, NM 88002

Office of the Test Director
Joint Services EO GW CM Test Program
ATTN: DRXDE-TD (Mr. Weldon Findley)
White Sands Missile Range, NM 88002

Commander
US Army White Sands Missile Range
ATTN: STEWS-PT-AL (Laurel B. Saunders)
White Sands Missile Range, NM 88002

Commander
US Army R&D Coordinator
US Embassy - Bonn
Box 165
APO New York 09080

Grumman Aerospace Corporation
Research Department - MS A08-35
ATTN: John E. A. Selby
Bethpage, NY 11714

Rome Air Development Center
ATTN: Documents Library
TSLD (Bette Smith)
Griffiss AFB, NY 13441

Dr. Roberto Vaglio-Laurin
Faculty of Arts and Science
Dept. of Applied Science
26-36 Stuyvesant Street
New York, NY 10003

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (Mr. Harold Geltmacher)
Wright-Patterson AFB, OH 45433

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (CPT William C. Smith)
Wright-Patterson AFB, OH 45433

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R (CPT James M. Watson)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CD-MS
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: NOAA Liaison Officer
(CDR Jeffrey G. Carlen)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
Morris Swett Library
ATTN: Reference Librarian
Fort Sill, OK 73503

Commander
Naval Air Development Center
ATTN: Code 301 (Mr. George F. Eck)
Warminster, PA 18974

The University of Texas at El Paso
Electrical Engineering Department
ATTN: Dr. Joseph H. Pierluissi
El Paso, TX 79968

Commandant
US Army Air Defense School
ATTN: ATSA-CD-SC-A (CPT Charles T. Thorn)
Fort Bliss, TX 79916

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-OP-Q (CPT Henry C. Cobb, Jr.)
Fort Hood, TX 76544

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-SCI (Dr. Darrell W. Collier)
Fort Hood, TX 76544

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-L
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-M (Mr. Paul E. Carlson)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. John Trethewey)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. William Peterson)
Dugway, UT 84022

Defense Documentation Center
ATTN: DDC-TCA
Cameron Station Bldg 5
Alexandria, VA 22314
12

Ballistic Missile Defense Program Office
ATTN: DACS-BMT (Colonel Harry F. Ennis)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Technical Information Center
ATTN: DDA-2 (Mr. James E. Shafer)
Cameron Station, Bldg 5
Alexandria, VA 22314

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCBSI-EE (Mr. Albert Giambalvo)
5001 Eisenhower Avenue
Alexandria, VA 22333

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCLDC (Mr. James Bender)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Advanced Rsch Projects Agency
ATTN: Steve Zakanyez
1400 Wilson Blvd
Arlington, VA 22209

Defense Advanced Rsch Projects Agency
ATTN: Dr. James Tegnelia
1400 Wilson Blvd
Arlington, VA 22209

Institute for Defense Analyses
ATTN: Mr. Lucien M. Biberman
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Ernest Bauer
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Hans G. Wolfhard
400 Army-Navy Drive
Arlington, VA 22202

System Planning Corporation
ATTN: Mr. Daniel Friedman
1500 Wilson Boulevard
Arlington, VA 22209

System Planning Corporation
ATTN: COL Hank Shelton
1500 Wilson Boulevard
Arlington, VA 22209

US Army Intelligence & Security Command
ATTN: Edwin Speakman, Scientific Advisor
Arlington Hall Station
Arlington, VA 22212

Commander
US Army Operational Test
& Evaluation Agency
ATTN: CSTE-ED (Mr. Floyd I. Hill)
5600 Columbia Pike
Falls Church, VA 22041

Commander and Director
US Army Engineer Topographic Laboratories
ATTN: ETL-GS-A (Mr. Thomas Neidringhaus)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Rudolf G. Buser)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Robert S. Rodhe)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Joseph R. Moulton)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Luanne P. Obert)
Fort Belvoir, VA 22060

Director
US Army Night Vision
& Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Thomas W. Cassidy)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Richard J. Bergemann)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Dr. James A. Ratches)
Fort Belvoir, VA 22060

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN-M
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-F-A (Mr. Chris O'Connor, Jr.)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-IE-R (Mr. David M. Ingram)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-M-I/ATCD-M-A
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATDOC-TA (Dr. Marvin P. Pastel)
Fort Monroe, VA 23651

Department of the Air Force
OL-I, AWS
Fort Monroe, VA 23651

Department of the Air Force
HQS 5 Weather Wing (MAC)
ATTN: 5 WW/DN
Langley Air Force Base, VA 23655

Commander
US Army INSCOM/Quest Research Corporation
ATTN: Mr. Donald Wilmot
6845 Elm Street, Suite 407
McLean, VA 22101

General Research Corporation
ATTN: Dr. Ralph Zirkind
7655 Old Springhouse Road
McLean, VA 22102

Science Applications, Inc.
8400 Westpark Drive
ATTN: Dr. John E. Cockayne
McLean, VA 22102

US Army Nuclear & Chemical Agency
ATTN: MONA-WE (Dr. John A. Berberet)
7500 Backlick Road, Bldg 2073
Springfield, VA 22150

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-EA (Mr. Douglas Harkleroad)
Vint Hill Farms Station
Warrenton, VA 22186

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-OS (Dr. Royal H. Burkhardt)
Vint Hill Farms Station
Warrenton, VA 22186

Commander
US Army Cold Regions Test Center
ATTN: STECR-TD (Mr. Jerold Barger)
APO Seattle, WA 98733

HQDA (SAUS-OR/Hunter M. Woodall, Jr./
Dr. Herbert K. Fallin)
Rm 2E 614, Pentagon
Washington, DC 20301

COL Elbert W. Friday, Jr.
OUSDRE
Rm 3D 129, Pentagon
Washington, DC 20301

Defense Communications Agency
Technical Library Center
Code 222
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: Technical Library (Mrs. Betty Fox)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: RAAE (Dr. Carl Fitz)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: SPAS (Mr. Donald J. Kohler)
Washington, DC 20305

Defense Intelligence Agency
ATTN: DT/AC (LTC Robert Poplawski)
Washington, DC 20301

HQDA (DAMA-ARZ-D/Dr. Verderame)
Washington, DC 20310

HQDA (DAMI-ISP/Mr. Beck)
Washington, DC 20310

Department of the Army
Deputy Chief of Staff for
Operations and Plans
ATTN: DAMO-RQ
Washington, DC 20310

Department of the Army
Director of Telecommunications and
Command and Control
ATTN: DAMO-TCZ
Washington, DC 20310

Department of the Army
Assistant Chief of Staff for Intelligence
ATTN: DAMI-TS
Washington, DC 20310

HQDA (DAEN-RDM/Dr. de Percin)
Casimir Pulaski Building
20 Massachusetts Avenue
Room 6203
Washington, DC 20314

National Science Foundation
Division of Atmospheric Sciences
ATTN: Dr. Eugene W. Bierly
1800 G. Street, N.W.
Washington, DC 20550

Director
Naval Research Laboratory
ATTN: Code 4320 (Dr. Lothar H. Ruhnke)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6009 (Dr. John MacCallum, Jr.)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6530 (Mr. Raymond A. Patten)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6533 (Dr. James A. Dowling)
Washington, DC 20375

ATMOSPHERIC SCIENCES RESEARCH REPORTS

1. Lindberg, J. D. "An Improvement to a Method for Measuring the Absorption Coefficient of Atmospheric Dust and other Strongly Absorbing Powders," ECOM-5565, July 1975.

2. Avara, Elton P., "Mesoscale Wind Shears Derived from Thermal Winds," ECOM-5566, July 1975.

3. Gomez, Richard B., and Joseph H. Pierluissi, "Incomplete Gamma Function Approximation for King's Strong-Line Transmittance Model," ECOM-5567, July 1975.

4. Blanco, A. J., and B. F. Engebos, "Ballistic Wind Weighting Functions for Tank Projectiles," ECOM-5568, August 1975.

5. Taylor, Fredrick J., Jack Smith, and Thomas H. Pries, "Crosswind Measurements through Pattern Recognition Techniques," ECOM-5569, July 1975.

6. Walters, D. L., "Crosswind Weighting Functions for Direct-Fire Projectiles," ECOM-5570, August 1975.

7. Duncan, Louis D., "An Improved Algorithm for the Iterated Minimal Information Solution for Remote Sounding of Temperature," ECOM-5571, August 1975.

8. Robbiani, Raymond L., "Tactical Field Demonstration of Mobile Weather Radar Set AN/TPS-41 at Fort Rucker, Alabama," ECOM-5572, August 1975.

9. Miers, B., G. Blackman, D. Langer, and N. Lorimier, "Analysis of SMS/GOES Film Data," ECOM-5573, September 1975.

10. Manquero, Carlos, Louis Duncan, and Rufus Bruce, "An Indication from Satellite Measurements of Atmospheric CO_2 Variability," ECOM-5574, September 1975.

11. Petracca, Carmine, and James D. Lindberg, "Installation and Operation of an Atmospheric Particulate Collector," ECOM-5575, September 1975.

12. Avara, Elton P., and George Alexander, "Empirical Investigation of Three Iterative Methods for Inverting the Radiative Transfer Equation," ECOM-5576, October 1975.

13. Alexander, George D., "A Digital Data Acquisition Interface for the SMS Direct Readout Ground Station - Concept and Preliminary Design," ECOM-5577, October 1975.

14. Cantor, Israel, "Enhancement of Point Source Thermal Radiation Under Clouds in a Nonattenuating Medium," ECOM-5578, October 1975.

15. Norton, Colburn, and Glenn Hoidale, "The Diurnal Variation of Mixing Height by Month over White Sands Missile Range, NM," ECOM-5579, November 1975.

16. Avara, Elton P., "On the Spectrum Analysis of Binary Data," ECOM-5580, November 1975.

17. Taylor, Fredrick J., Thomas H. Pries, and Chao-Huan Huang, "Optimal Wind Velocity Estimation," ECOM-5581, December 1975.

18. Avara, Elton P., "Some Effects of Autocorrelated and Cross-Correlated Noise on the Analysis of Variance," ECOM-5582, December 1975.

19. Gillespie, Patti S., R. L. Armstrong, and Kenneth O. White, "The Spectral Characteristics and Atmospheric CO_2 Absorption of the Ho^{+3}:YLF Laser at 2.05μm," ECOM-5583, December 1975.

20. Novlan, David J., "An Empirical Method of Forecasting Thunderstorms for the White Sands Missile Range," ECOM-5584, February 1976.

21. Avara, Elton P., "Randomization Effects in Hypothesis Testing with Autocorrelated Noise," ECOM-5585, February 1976.

22. Watkins, Wendell R., "Improvements in Long Path Absorption Cell Measurement," ECOM-5586, March 1976.

23. Thomas, Joe, George D. Alexander, and Marvin Dubbin, "SATTEL - An Army Dedicated Meteorological Telemetry System," ECOM-5587, March 1976.

24. Kennedy, Bruce W., and Delbert Bynum, "Army User Test Program for the RDT&E-XM-75 Meteorological Rocket," ECOM-5588, April 1976.

25. Barnett, Kenneth M., "A Description of the Artillery Meteorological Comparisons at White Sands Missile Range, October 1974 - December 1974 ('PASS' - Prototype Artillery [Meteorological] Subsystem)," ECOM-5589, April 1976.

26. Miller, Walter B., "Preliminary Analysis of Fall-of-Shot From Project 'PASS'," ECOM-5590, April 1976.

27. Avara, Elton P., "Error Analysis of Minimum Information and Smith's Direct Methods for Inverting the Radiative Transfer Equation," ECOM-5591, April 1976.

28. Yee, Young P., James D. Horn, and George Alexander, "Synoptic Thermal Wind Calculations from Radiosonde Observations Over the Southwestern United States," ECOM-5592, May 1976.

29. Duncan, Louis D., and Mary Ann Seagraves, "Applications of Empirical Corrections to NOAA-4 VTPR Observations," ECOM-5593, May 1976.

30. Miers, Bruce T., and Steve Weaver, "Applications of Meteorological Satellite Data to Weather Sensitive Army Operations," ECOM-5594, May 1976.

31. Sharenow, Moses, "Redesign and Improvement of Balloon ML-566," ECOM-5595, June 1976.

32. Hansen, Frank V., "The Depth of the Surface Boundary Layer," ECOM-5596, June 1976.

33. Pinnick, R. G., and E. B. Stenmark, "Response Calculations for a Commerical Light-Scattering Aerosol Counter," ECOM-5597, July 1976.

34. Mason, J., and G. B. Hoidale, "Visibility as an Estimator of Infrared Transmittance," ECOM-5598, July 1976.

35. Bruce, Rufus E., Louis D. Duncan, and Joseph H. Pierluissi, "Experimental Study of the Relationship Between Radiosonde Temperatures and Radiometric-Area Temperatures," ECOM-5599, August 1976.

36. Duncan, Louis D., "Stratospheric Wind Shear Computed from Satellite Thermal Sounder Measurements," ECOM-5800, September 1976.

37. Taylor, F., P. Mohan, P. Joseph, and T. Pries, "An All Digital Automated Wind Measurement System," ECOM-5801, September 1976.

38. Bruce, Charles, "Development of Spectrophones for CW and Pulsed Radiation Sources," ECOM-5802, September 1976.

39. Duncan, Louis D., and Mary Ann Seagraves, "Another Method for Estimating Clear Column Radiances," ECOM-5803, October 1976.

40. Blanco, Abel J., and Larry E. Taylor, "Artillery Meteorological Analysis of Project Pass," ECOM-5804, October 1976.

41. Miller, Walter, and Bernard Engebos, "A Mathematical Structure for Refinement of Sound Ranging Estimates," ECOM-5805, November 1976.

42. Gillespie, James B., and James D. Lindberg, "A Method to Obtain Diffuse Reflectance Measurements from 1.0 and 3.0μm Using a Cary 17I Spectrophotometer," ECOM-5806, November 1976.

43. Rubio, Roberto, and Robert O. Olsen, "A Study of the Effects of Temperature Variations on Radio Wave Absorption," ECOM-5807, November 1976.

44. Ballard, Harold N., "Temperature Measurements in the Stratosphere from Balloon-Borne Instrument Platforms, 1968-1975," ECOM-5808, December 1976.

45. Monahan, H. H., "An Approach to the Short-Range Prediction of Early Morning Radiation Fog," ECOM-5809, January 1977.

46. Engebos, Bernard Francis, "Introduction to Multiple State Multiple Action Decision Theory and Its Relation to Mixing Structures," ECOM-5810, January 1977.

47. Low, Richard D. H., "Effects of Cloud Particles on Remote Sensing from Space in the 10-Micrometer Infrared Region," ECOM-5811, January 1977.

48. Bonner, Robert S., and R. Newton, "Application of the AN/GVS-5 Laser Rangefinder to Cloud Base Height Measurements," ECOM-5812, February 1977.

49. Rubio, Roberto, "Lidar Detection of Subvisible Reentry Vehicle Erosive Atmospheric Material," ECOM-5813, March 1977.

50. Low, Richard D. H., and J. D. Horn, "Mesoscale Determination of Cloud-Top Height: Problems and Solutions," ECOM-5814, March 1977.

51. Duncan, Louis D., and Mary Ann Seagraves, "Evaluation of the NOAA-4 VTPR Thermal Winds for Nuclear Fallout Predictions," ECOM-5815, March 1977.

52. Randhawa, Jagir S., M. Izquierdo, Carlos McDonald, and Zvi Salpeter, "Stratospheric Ozone Density as Measured by a Chemiluminescent Sensor During the Stratcom VI-A Flight," ECOM-5816, April 1977.

53. Rubio, Roberto, and Mike Izquierdo, "Measurements of Net Atmospheric Irradiance in the 0.7- to 2.8-Micrometer Infrared Region," ECOM-5817, May 1977.

54. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, Consultant for Chemical Kinetics, "Calculation of Selected Atmospheric Composition Parameters for the Mid-Latitude, September Stratosphere," ECOM-5818, May 1977.

55. Mitchell, J. D., R. S. Sagar, and R. O. Olsen, "Positive Ions in the Middle Atmosphere During Sunrise Conditions," ECOM-5819, May 1977.

56. White, Kenneth O., Wendell R. Watkins, Stuart A. Schleusener, and Ronald L. Johnson, "Solid-State Laser Wavelength Identification Using a Reference Absorber," ECOM-5820, June 1977.

57. Watkins, Wendell R., and Richard G. Dixon, "Automation of Long-Path Absorption Cell Measurements," ECOM-5821, June 1977.

58. Taylor, S. E., J. M. Davis, and J. B. Mason, "Analysis of Observed Soil Skin Moisture Effects on Reflectance," ECOM-5822, June 1977.

59. Duncan, Louis D., and Mary Ann Seagraves, "Fallout Predictions Computed from Satellite Derived Winds," ECOM-5823, June 1977.

60. Snider, D. E., D. G. Murcray, F. H. Murcray, and W. J. Williams, "Investigation of High-Altitude Enhanced Infrared Backround Emissions," (U), SECRET, ECOM-5824, June 1977.

51. Cubbin, Marvin H., and Dennis Hall, "Synchronous Meteorological Satellite Direct Readout Ground System Digital Video Electronics," ECOM-5825, June 1977.

62. Miller, W., and B. Engebos, "A Preliminary Analysis of Two Sound Ranging Algorithms," ECOM-5826, July 1977.

63. Kennedy, Bruce W., and James K. Luers, "Ballistic Sphere Techniques for Measuring Atmospheric Parameters," ECOM-5827, July 1977.

64. Duncan, Louis D., "Zenith Angle Variation of Satellite Thermal Sounder Measurements," ECOM-5828, August 1977.

65. Hansen, Frank V., "The Critical Richardson Number," ECOM-5829, September 1977.

66. Ballard, Harold N., and Frank P. Hudson (Compilers), "Stratospheric Composition Balloon-Borne Experiment," ECOM-5830, October 1977.

67. Barr, William C., and Arnold C. Peterson, "Wind Measuring Accuracy Test of Meteorological Systems," ECOM-5831, November 1977.

68. Ethridge, G. A., and F. V. Hansen, "Atmospheric Diffusion: Similarity Theory and Empirical Derivations for Use in Boundary Layer Diffusion Problems," ECOM-5832, November 1977.

69. Low, Richard D. H., "The Internal Cloud Radiation Field and a Technique for Determining Cloud Blackness," ECOM-5833, December 1977.

70. Watkins, Wendell R., Kenneth O. White, Charles W. Bruce, Donald L. Walters, and James D. Lindberg, "Measurements Required for Prediction of High Energy Laser Transmission," ECOM-5834, December 1977.

71. Rubio, Robert, "Investigation of Abrupt Decreases in Atmospherically Backscattered Laser Energy," ECOM-5835, December 1977.

72. Monahan, H. H., and R. M. Cionco, "An Interpretative Review of Existing Capabilities for Measuring and Forecasting Selected Weather Variables (Emphasizing Remote Means)," ASL-TR-0001, January 1978.

73. Heaps, Melvin G., "The 1979 Solar Eclipse and Validation of D-Region Models," ASL-TR-0002, March 1978.

74. Jennings, S. G., and J. B. Gillespie, "M.I.E. Theory Sensitivity Studies - The Effects of Aerosol Complex Refractive Index and Size Distribution Variations on Extinction and Absorption Coefficients, Part II: Analysis of the Computational Results," ASL-TR-0003, March 1978.

75. White, Kenneth O., et al, "Water Vapor Continuum Absorption in the 3.5μm to 4.0μm Region," ASL-TR-0004, March 1978.

76. Olsen, Robert O., and Bruce W. Kennedy, "ABRES Pretest Atmospheric Measurements," ASL-TR-0005, April 1978.

77. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, "Calculation of Atmospheric Composition in the High Latitude September Stratosphere," ASL-TR-0006, May 1978.

78. Watkins, Wendell R., et al, "Water Vapor Absorption Coefficients at HF Laser Wavelengths," ASL-TR-0007, May 1978.

79. Hansen, Frank V., "The Growth and Prediction of Nocturnal Inversions," ASL-TR-0008, May 1978.

80. Samuel, Christine, Charles Bruce, and Ralph Brewer, "Spectrophone Analysis of Gas Samples Obtained at Field Site," ASL-TR-0009, June 1978.

81. Pinnick, R. G., et al., "Vertical Structure in Atmospheric Fog and Haze and its Effects on IR Extinction," ASL-TR-0010, July 1978.

82. Low, Richard D. H., Louis D. Duncan, and Richard B. Gomez, "The Microphysical Basis of Fog Optical Characterization," ASL-TR-0011, August 1978.

83. Heaps, Melvin G., "The Effect of a Solar Proton Event on the Minor Neutral Constituents of the Summer Polar Mesosphere," ASL-TR-0012, August 1978.

84. Mason, James B., "Light Attenuation in Falling Snow," ASL-TR-0013, August 1978.

85. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' Meteorological Application," ASL-TR-0014, September 1978.

86. Heaps, M. G., and F. E. Niles, "Modeling of Ion Chemistry of the D-Region: A Case Study Based Upon the 1966 Total Solar Eclipse," ASL-TR-0015, September 1978.

87. Jennings, S. G., and R. G. Pinnick, "Effects of Particulate Complex Refractive Index and Particle Size Distribution Variations on Atmospheric Extinction and Absorption for Visible Through Middle-Infrared Wavelengths," ASL-TR-0016, September 1978.

88. Watkins, Wendell R., Kenneth O. White, Lanny R. Bower, and Brian Z. Sojka, "Pressure Dependence of the Water Vapor Continuum Absorption in the 3.5- to 4.0-Micrometer Region," ASL-TR-0017, September 1978.

89. Miller, W. B., and B. F. Engebos, "Behavior of Four Sound Ranging Techniques in an Idealized Physical Environment," ASL-TR-0018, September 1978.

90. Gomez, Richard G., "Effectiveness Studies of the CBU-88/B Bomb, Cluster, Smoke Weapon," (U), CONFIDENTIAL ASL-TR-0019, September 1978.

91. Miller, August, Richard C. Shirkey, and Mary Ann Seagraves, "Calculation of Thermal Emission from Aerosols Using the Doubling Technique," ASL-TR-0020, November 1978.

92. Lindberg, James D., et al, "Measured Effects of Battlefield Dust and Smoke on Visible, Infrared, and Millimeter Wavelengths Propagation: A Preliminary Report on Dusty Infrared Test-I (DIRT-I)," ASL-TR-0021, January 1979.

93. Kennedy, Bruce W., Arthur Kinghorn, and B. R. Hixon, "Engineering Flight Tests of Range Meteorological Sounding System Radiosonde," ASL-TR-0022, February 1979.

94. Rubio, Roberto, and Don Hoock, "Microwave Effective Earth Radius Factor Variability at Wiesbaden and Balboa," ASL-TR-0023, February 1979.

95. Low, Richard D. H., "A Theoretical Investigation of Cloud/Fog Optical Properties and Their Spectral Correlations, "ASL-TR-0024, February 1979.

96. Pinnick, R. G., and H. J. Auvermann, "Response Characteristics of Knollenberg Light-Scattering Aerosol Counters," ASL-TR-0025, February 1979.

97. Heaps, Melvin G., Robert O. Olsen, and Warren W. Berning, "Solar Eclipse 1979, Atmospheric Sciences Laboratory Program Overview," ASL-TR-0026, February 1979.

98. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' GR-8 Sound Ranging Data," ASL-TR-0027, March 1979.

99. Kennedy, Bruce W., and Jose M. Serna, "Meteorological Rocket Network System Reliability," ASL-TR-0028, March 1979.

100. Swingle, Donald M., "Effects of Arrival Time Errors in Weighted Range Equation Solutions for Linear Base Sound Ranging," ASL-TR-0029, April 1979.

101. Umstead, Robert K., Ricardo Pena, and Frank V. Hansen, "KWIK: An Algorithm for Calculating Munition Expenditures for Smoke Screening/Obscuration in Tactical Situations," ASL-TR-0030, April 1979.

102. D'Arcy, Edward M., "Accuracy Validation of the Modified Nike Hercules Radar," ASL-TR-0031, May 1979.

103. Rodriguez, Ruben, "Evaluation of the Passive Remote Crosswind Sensor," ASL-TR-0032, May 1979.

104. Barber, T. L., and R. Rodriguez, "Transit Time Lidar Measurement of Near-Surface Winds in the Atmosphere," ASL-TR-0033, May 1979.

105. Low, Richard D. H., Louis D. Duncan, and Y. Y. Roger R. Hsiao, "Microphysical and Optical Properties of California Coastal Fogs at Fort Ord," ASL-TR-0034, June 1979.

106. Rodriguez, Ruben, and William J. Vechione, "Evaluation of the Saturation Resistant Crosswind Sensor," ASL-TR-0035, July 1979.

107. Ohmstede, William D., "The Dynamics of Material Layers," ASL-TR-0036, July 1979.

108. Pinnick, R. G., S. G. Jennings, Petr Chylek, and H. J. Auvermann, "Relationships between IR Extinction Absorption, and Liquid Water Content of Fogs," ASL-TR-0037, August 1979.

109. Rodriguez, Ruben, and William J. Vechione, "Performance Evaluation of the Optical Crosswind Profiler," ASL-TR-0038, August 1979.

110. Miers, Bruce T., "Precipitation Estimation Using Satellite Data," ASL-TR-0039, September 1979.

111. Dickson, David H., and Charles M. Sonnenschein, "Helicopter Remote Wind Sensor System Description," ASL-TR-0040, September 1979.

112. Heaps, Melvin G., and Joseph M. Heimerl, "Validation of the Dairchem Code, I: Quiet Midlatitude Conditions," ASL-TR-0041, September 1979.

113. Bonner, Robert S., and William J. Lentz, "The Visioceilometer: A Portable Cloud Height and Visibility Indicator," ASL-TR-0042, October 1979.

114. Cohn, Stephen L., "The Role of Atmospheric Sulfates in Battlefield Obscurations," ASL-TR-0043, October 1979.

115. Fawbush, E. J., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF), White Sands Missile Range, New Mexico, Part I, 24 March to 8 April 1977," ASL-TR-0044, November 1979.

116. Barber, Ted L., "Short-Time Mass Variation in Natural Atmospheric Dust," ASL-TR-0045, November 1979.

117. Low, Richard D. H., "Fog Evolution in the Visible and Infrared Spectral Regions and its Meaning in Optical Modeling," ASL-TR-0046, December 1979.

118. Duncan, Louis D., et al, "The Electro-Optical Systems Atmospheric Effects Library, Volume I: Technical Documentation," ASL-TR-0047, December 1979.

119. Shirkey, R. C., et al, "Interim E-O SAEL, Volume II, Users Manual," ASL-TR-0048, December 1979.

120. Kobayashi, H. K., "Atmospheric Effects on Millimeter Radio Waves," ASL-TR-0049, January 1980.

121. Seagraves, Mary Ann, and Louis D. Duncan, "An Analysis of Transmittances Measured Through Battlefield Dust Clouds," ASL-TR-0050, February 1980.

122. Dickson, David H., and Jon E. Ottesen, "Helicopter Remote Wind Sensor Flight Test," ASL-TR-0051, February 1980.

123. Pinnick, R. G., and S. G. Jennings, "Relationships Between Radiative Properties and Mass Content of Phosphoric Acid, HC, Petroleum Oil, and Sulfuric Acid Military Smokes," ASL-TR-0052, April 1980.

124. Hinds, B. D., and J. B. Gillespie, "Optical Characterization of Atmospheric Particulates on San Nicolas Island, California," ASL-TR-0053, April 1980.

125. Miers, Bruce T., "Precipitation Estimation for Military Hydrology," ASL-TR-0054, April 1980.

126. Stenmark, Ernest B., "Objective Quality Control of Artillery Computer Meteorological Messages," ASL-TR-0055, April 1980.

127. Duncan, Louis D., and Richard D. H. Low, "Bimodal Size Distribution Models for Fogs at Meppen, Germany," ASL-TR-0056, April 1980.

128. Olsen, Robert O., and Jagir S. Randhawa, "The Influence of Atmospheric Dynamics on Ozone and Temperature Structure," ASL-TR-0057, May 1980.

129. Kennedy, Bruce W., et al, "Dusty Infrared Test-II (DIRT-II) Program," ASL-TR-0058, May 1980.

130. Heaps, Melvin G., Robert O. Olsen, Warren Berning, John Cross, and Arthur Gilcrease, "1979 Solar Eclipse, Part I - Atmospheric Sciences Laboratory Field Program Summary," ASL-TR-0059, May 1980

131. Miller, Walter B., "User's Guide for Passive Target Acquisition Program Two (PTAP-2)," ASL-TR-0060, June 1980.

132. Holt, E. H., editor, "Atmospheric Data Requirements for Battlefield Obscuration Applications," ASL-TR-0061, June 1980.

133. Shirkey, Richard C., August Miller, George H. Goedecke, and Yugal Behl, "Single Scattering Code AGAUSX: Theory, Applications, Comparisons, and Listing," ASL-TR-0062, July 1980.

134. Sojka, Brian Z., and Kenneth O. White, "Evaluation of Specialized Photoacoustic Absorption Chambers for Near-Millimeter Wave (NMMW) Propagation Measurements," ASL-TR-0063, August 1980.

135. Bruce, Charles W., Young Paul Yee, and S. G. Jennings, "In Situ Measurement of the Ratio of Aerosol Absorption to Extinction Coefficient," ASL-TR-0064, August 1980.

136. Yee, Young Paul, Charles W. Bruce, and Ralph J. Brewer, "Gaseous/Particulate Absorption Studies at WSMR using Laser Sourced Spectrophones," ASL-TR-0065, June 1980.

137. Lindberg, James D., Radon B. Loveland, Melvin Heaps, James B. Gillespie, and Andrew F. Lewis, "Battlefield Dust and Atmospheric Characterization Measurements During West German Summertime Conditions in Support of Grafenwohr Tests," ASL-TR-0066, September 1980.

138. Vechione, W. J., "Evaluation of the Environmental Instruments, Incorporated Series 200 Dual Component Wind Set," ASL-TR-0067, September 1980.

139. Bruce, C. W., Y. P. Yee, B. D. Hinds, R. G. Pinnick, R. J. Brewer, and J. Minjares, "Initial Field Measurements of Atmospheric Absorption at 9μm to 11μm Wavelengths," ASL-TR-0068, October 1980.

140. Heaps, M. G., R. O. Olsen, K. D. Baker, D. A. Burt, L. C. Howlett, L. L. Jensen, E. F. Pound, and G. D. Allred, "1979 Solar Eclipse: Part II Initial Results for Ionization Sources, Electron Density, and Minor Neutral Constituents," ASL-TR-0069, October 1980.

141. Low, Richard D. H., "One-Dimensional Cloud Microphysical Models for Central Europe and their Optical Properties," ASL-TR-0070, October 1980.

142. Duncan, Louis D., James D. Lindberg, and Radon B. Loveland, "An Empirical Model of the Vertical Structure of German Fogs," ASL-TR-0071, November 1980.

143. Duncan, Louis D., "EOSAEL 80, Volume I, Technical Documentation," ASL-TR-0072, January 1981.

144. Shirkey, R. C., and S. G. O'Brien, "EOSAEL 80, Volume II, Users Manual," ASL-TR-0073, January 1981.

145. Bruce, C. W., "Characterization of Aerosol Nonlinear Effects on a High-Power CO_2 Laser Beam," ASL-TR-0074, February 1981.

146. Duncan, Louis D., and James D. Lindberg, "Air Mass Considerations in Fog Optical Modeling," ASL-TR-0075, February 1981.

147. Kunkel, Kenneth E., "Evaluation of a Tethered Kite Anemometer," ASL-TR-0076, February 1981.

148. Kunkel, K. E., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF) White Sands Missile Range, New Mexico, August 1977 to October 1978, Part II, Optical Turbulence, Wind, Water Vapor Pressure, Temperature," ASL-TR-0077, February 1981.

149. Miers, Bruce T., "Weather Scenarios for Central Germany," ASL-TR-0078, February 1981.

150. Cogan, James L., "Sensitivity Analysis of a Mesoscale Moisture Model," ASL-TR-0079, March 1981.

151. Brewer, R. J., C. W. Bruce, and J. L. Mater, "Optoacoustic Spectroscopy of C_2H_4 at the 9μm and 10μm $C^{12}O_2^{16}$ Laser Wavelengths," ASL-TR-0080, March 1981.

152. Swingle, Donald M., "Reducible Errors in the Artillery Sound Ranging Solution, Part I: The Curvature Correction" (U), SECRET, ASL-TR-0081, April 1981.

153. Miller, Walter B., "The Existence and Implications of a Fundamental System of Linear Equations in Sound Ranging" (U), SECRET, ASL-TR-0082, April 1981.

154. Bruce, Dorothy, Charles W. Bruce, and Young Paul Yee, "Experimentally Determined Relationship Between Extinction and Liquid Water Content," ASL-TR-0083, April 1981.

155. Seagraves, Mary Ann, "Visible and Infrared Obscuration Effects of Ice Fog," ASL-TR-0084, May 1981.

156. Watkins, Wendell R., and Kenneth O. White, "Wedge Absorption Remote Sensor," ASL-TR-0085, May 1981.

157. Watkins, Wendell R., Kenneth O. White, and Laura J. Crow, "Turbulence Effects on Open Air Multipaths," ASL-TR-0086, May 1981.

158. Blanco, Abel J., "Extending Application of the Artillery Computer Meteorological Message," ASL-TR-0087, May 1981.

159. Heaps, M. G., D. W. Hoock, R. O. Olsen, B. F. Engebos, and R. Rubio, "High Frequency Position Location: An Assessment of Limitations and Potential Improvements," ASL-TR-0088, May 1981.

160. Watkins, Wendell R., and Kenneth O. White, "Laboratory Facility for Measurement of Hot Gaseous Plume Radiative Transfer," ASL-TR-0089, June 1981.

161. Heaps, M. G., "Dust Cloud Models: Sensitivity of Calculated Transmittances to Variations in Input Parameters," ASL-TR-0090, June 1981.

162. Seagraves, Mary Ann, "Some Optical Properties of Blowing Snow," ASL-TR-0091, June 1981.

163. Kobayashi, Herbert K., "Effect of Hail, Snow, and Melting Hydrometeors on Millimeter Radio Waves," ASL-TR-0092, July 1981.

164. Cogan, James L., "Techniques for the Computation of Wind, Ceiling, and Extinction Coefficient Using Currently Acquired RPV Data," ASL-TR-0093, July 1981.

165. Miller, Walter B., and Bernard F. Engebos, "On the Possibility of Improved Estimates for Effective Wind and Temperature," (U), SECRET, ASL-TR-0094, August 1981.

166. Heaps, Melvin G., "The Effect of Ionospheric Variability on the Accuracy of High Frequency Position Location," ASL-TR-0095, August 1981.

167. Sutherland, Robert A., Donald W. Hoock, and Richard B. Gomez, "An Objective Summary of US Army Electro-Optical Modeling and Field Testing in an Obscuring Environment," ASL-TR-0096, October 1981.

168. Pinnick, R. G., et al, "Backscatter and Extinction in Water Clouds," ASL-TR-0097, October 1981.

169. Cole, Henry P., and Melvin G. Heaps, "Properties of Dust as an Electron and Ion Attachment Site for Use in D Region Ion Chemistry," ASL-TR-0098, October 1981.

170. Spellicy, Robert L., Laura J. Crow, and Kenneth O. White, "Water Vapor Absorption Coefficients at HF Laser Wavelengths Part II: Development of the Measurement System and Measurements at Simulated Altitudes to 10 KM," ASL-TR-0099, November 1981.

171. Cohn, Stephen L., "Transport and Diffusion Solutions for Obscuration Using the XM-825 Smoke Munition," ASL-TR-0100, November 1981.

172. Pinnick, R. G., D. M. Garvey, and L. D. Duncan, "Calibration of Knollenberg FSSP Light-Scattering Counters for Measurement of Cloud Droplets," ASL-TR-0101, December 1981.

173. Cohn, Stephen L. and Ricardo Pena, "Munition Expenditure Model Verification: KWIK Phase I," ASL-TR-0102, December 1981.

174. Blackman, George R., "Cloud Geometry Analysis of the Smoke Week III Obscuration Trials," ASL-TR-0103, January 1982.

175. Sutherland, R. A., and D. W. Hoock, "An Improved Smoke Obscuration Model ACT II: Part 1 Theory," ASL-TR-0104, January 1982.

176. Lentz, W. J., "The Visioceilometer: A Portable Visibility and Cloud Ceiling Height Lidar," ASL-TR-0105, January 1982

177. Kennedy, Bruce W., "An Analysis of Craters Produced by Artillery Munitions During Dusty Infrared Test Series," ASL-TR-0106, January 1982.

178. Steinhoff, R. G., "Program Listings for EOSAEL 80-B and Ancillary Codes AGAUS and FLASH," ASL-TR-0107, February 1982.

DATE

ILME